

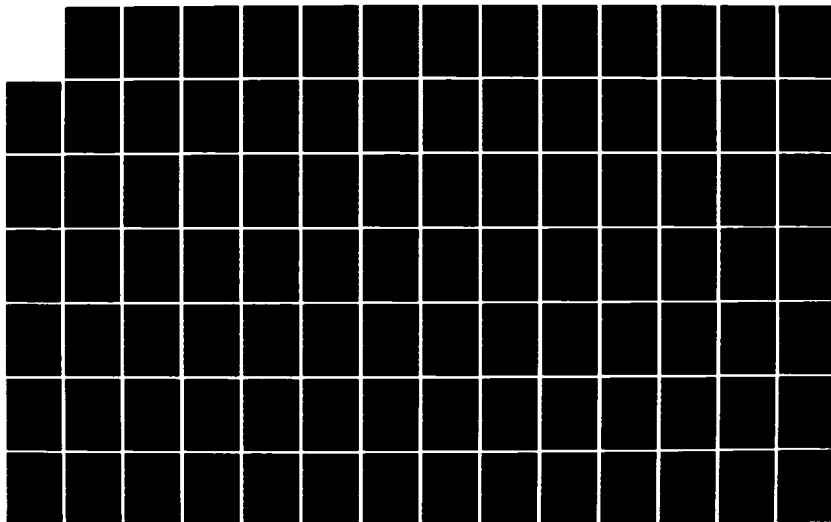
AD-A132 589

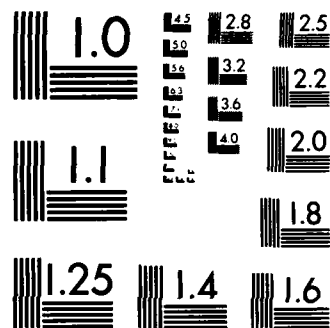
FORECASTING OUTCOMES OF MULTILATERAL NEGOTIATIONS:
COMPUTER PROGRAMS VOLU. (U) CENTER FOR NAVAL ANALYSES
ALEXANDRIA VA INST OF NAVAL STUDIES. K W GOUDREAU
OCT 76 CRC-290-VOL-2 N00014-76-C-0001 F/G 9/2

1/2

UNCLASSIFIED

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

CRC 290 - Vol. II

FORECASTING OUTCOMES OF MULTILATERAL NEGOTIATIONS: COMPUTER PROGRAMS

Guide for Programmers

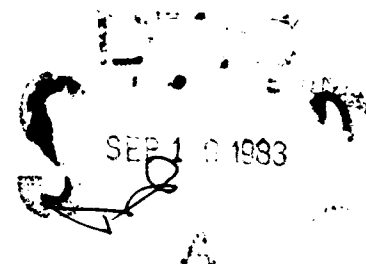
CENTER FOR NAVAL ANALYSES

1401 Wilson Boulevard
Arlington, Virginia 22209

Institute of Naval Studies

By: Karen W. Goudreau

October 1976



Approved for public release; distribution unlimited.

NTTC FILE COPY

Prepared for:

OFFICE OF NAVAL RESEARCH

Department of the Navy

Arlington, Virginia 22217

OFFICE OF THE CHIEF OF NAVAL OPERATIONS (Op96)

Department of the Navy

Washington, D.C. 20350

02 029002.00

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CRC 290 - Volume II	2. GOVT ACCESSION NO. AD-A132577	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Forecasting Outcomes of Multilateral Negotiations: Computer Programs - - Guide for Programmers		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Karen W. Goudreau		8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS Center for Naval Analyses 1401 Wilson Boulevard Arlington, Virginia 22209		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Department of the Navy Arlington, Virginia 22217		12. REPORT DATE October 1976
		13. NUMBER OF PAGES 112
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of the Chief of Naval Operations (Op96) Department of the Navy Washington, D.C. 20350		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES This Research Contribution does not necessarily represent the opinion of the Department of the Navy.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) computer programs, Fortran, Law of the Sea, multilateral negotiations, programmers guide		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The present Research Contribution is a companion publication to CNA Research Contribution 291, which describes the methodology developed and used by the Law of the Sea Study to forecast the outcomes of multilateral negotiations on issues in law of the sea. All of the mathematical models and most of the techniques were pro- grammed for the computer, and it is this system of computer programs that is treated in this Research Contribution. Volume I directs the prospective user in their exe- cution. Volume II describes their logic and dimensioned variables for the pro-		

20
grammer and specifies the changes that must be made throughout the system when any program is changed.

1401 Wilson Boulevard

Arlington, Virginia 22209

703/524-9400

An Equal Opportunity Employer

**Center
for
Naval
Analyses**

an affiliate of the
University of Rochester

9 December 1976

MEMORANDUM FOR DISTRIBUTION LIST

Subj: Center for Naval Analyses Research Contribution 290,
Volume II

Encl: (1) CRC 290, "Forecasting Outcomes of Multilateral
Negotiations: Computer Programs - Volume II:
Guide for Programmers," by Karen W. Goudreau,
October 1976

1. Enclosure (1) is forwarded as a matter of possible interest.
2. Research Contributions are distributed for their potential value in other studies and analyses. They do not necessarily represent the opinion of the Department of the Navy.

Herschel Kanter

HERSCHEL E. KANTER
Director
Institute of Naval Studies

DISTRIBUTION LIST:
National Defense University
Armed Forces Staff College
Department of Defense Computer Institute
U.S. Naval Academy
Nimitz Library, U.S. Naval Academy
Naval Postgraduate School
Naval War College
OpNav: Op-09BH, Op-96
Advanced Research Projects Agency
Institute for Defense Analyses
The Rand Corporation
National Archives, Machine Readable Branch

Accession For	
ADAMS	<input checked="" type="checkbox"/>
ADAMS	<input type="checkbox"/>
ADAMS	<input type="checkbox"/>
ADAMS	<input type="checkbox"/>
By _____	
Date _____	
Initials _____	

0110

A

TABLE OF CONTENTS

	Page
Preface	iii
Maintaining the data tapes	
Structure of the data tapes	1
Program CHECK	1
Program REMOVE	4
Program UPDATE	8
Maintaining the dictionary tape	
Structure of the dictionary tape	12
Program SEARCH	12
Program DTAPE	16
Program CODEBK	36
Organizing the data	
Program MATRIX	41
Program PTHEME	48
Program LETTER	55
Program PROFLE	59
Summarizing country positions	
Program SCORES	72
Program BIMED34	81
Program MBLOC	81
Forecasting outcomes	
Program VMODEL	87
Program MAXPAC	95
Program VALID	104
Appendix A - Programs affected by changes to dictionary and data tapes	A-1 - A-3

PREFACE

This volume describes the computer programs shown in figure 1. The system consists of 15 programs written in FORTRAN for operation on CNA's CDC 3800 computer and a standard statistical regression program, BIMED34. For simplicity, only the machine-readable output is shown in figure 1. Every program produces a printed output.

The description of each program consists of a statement of its purpose, a description of its logic, the definition of its dimensioned variables, a list of changes to be made to it as a result of changes to other programs and vice versa, and a listing of the program and subroutines. Whenever a program produces an output tape, its structure and format are also described. The appendix summarizes the effect of changes to the dictionary and data tapes on all the programs.

The reader should refer to volume I for a glossary of terms and a summary of the methodology, including the mathematical models that are the bases for certain programs. If a more detailed discussion of techniques and models is wanted, the reader should refer to CNA RC 291.

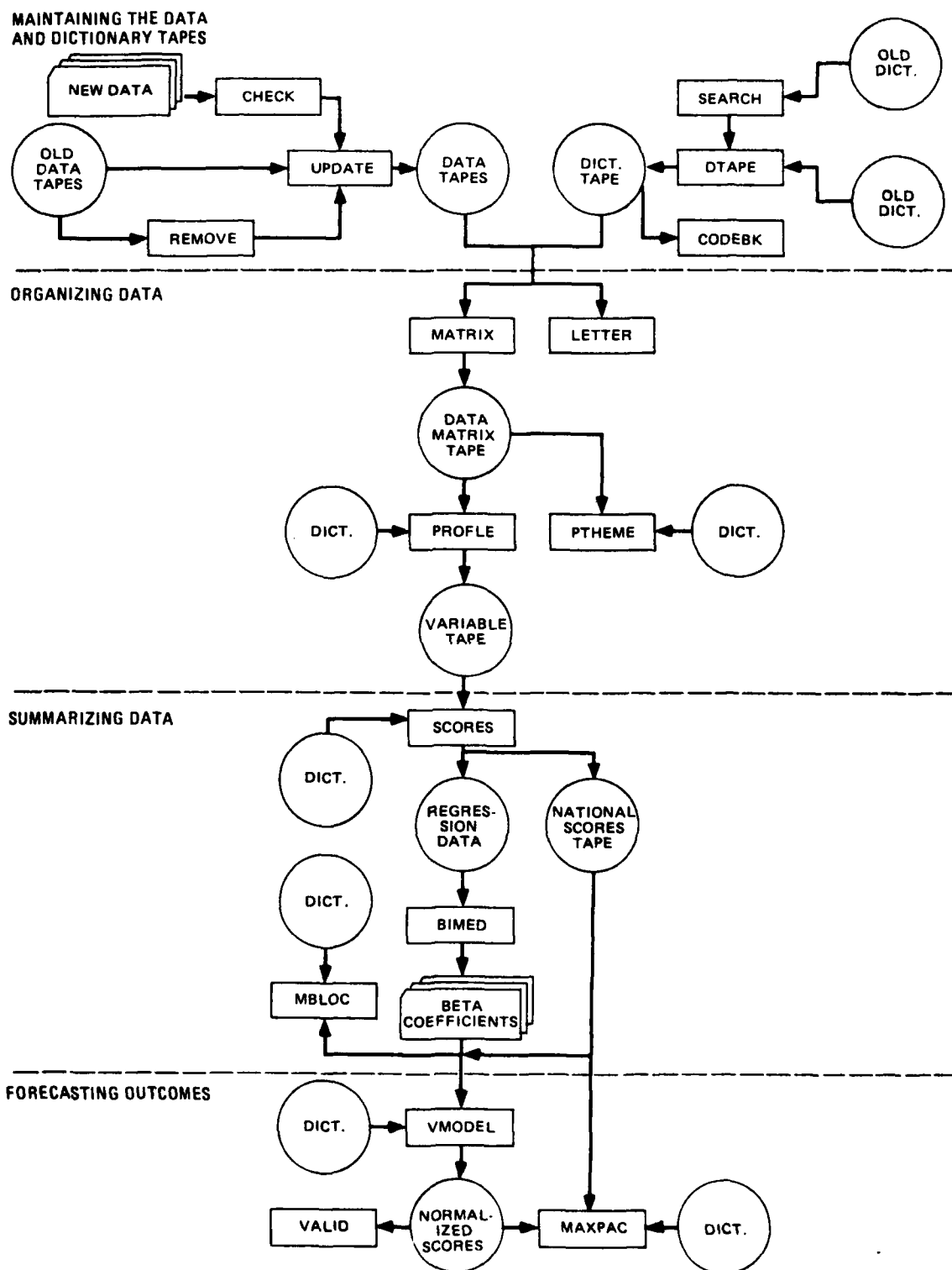


FIG. 1: FLOW CHART OF THE LOS SYSTEM OF PROGRAMS

MAINTAINING THE DATA TAPES

STRUCTURE OF THE DATA TAPES

There are three law of the sea data bases. The first, on two reels, contains all data that has been extracted from the primary source documents, the second contains all data from the secondary source documents, and the third contains all inactive data from both primary and secondary sources. The first two data bases are referred to as the active data base, and the third is referred to as the historical data base. For purposes of analysis only the active data base has been used.

The tapes are unblocked and 7-tracked. Each is organized by country number and, within country, numerically by theme number. The primary data tapes contain approximately 30,000 records, the secondary tape 12,000 records, and the historical tape 8,000 records.

A record is equivalent to an observation and is written in the following BCD format:

I3, 1X, 4I1, 2X, I3, 6X, I4, A1.

The eight fields contain: (1) country number, (2) regional flag, (3) repetition flag, (4) classification code, (5) reliability code, (6) document number, (7) theme number, and (8) letter associated with the theme number. For a detailed description of these entries, the reader should refer to appendix A of the users guide.

PROGRAM CHECK

Purpose

This program verifies that new observations have been accurately transcribed and keypunched. It produces a list of observations in the order in which they were extracted from the documents.

Method

The country names are read off the dictionary tape and stored in NACON. The input deck of observations is read a card at a time. A document header is produced whenever the document number changes and a country header is produced whenever the country number changes. Each observation is printed with its theme number and letter designation, if any.

Dimensioned Variable

NACON, dimensioned 149 by 2, contains a two field name for each country.

Operational Effects of System Changes

This program depends upon the current structure and format of a new observation (update card) and the country dictionary. If an update card is modified, format 2 and statement 10 may have to be changed. If the number of countries exceeds 149, MAXC must be increased and NACON redimensioned. If the country dictionary file is restructured or reformatted, format 1 may have to be changed.

Program Listing

The listing of program CHECK follows.

```

PROGRAM CHECK
DIMENSION NACCN(149,2)
DATA(NBLANK=1H )

C
C   THIS PROGRAM IS USED TO VERIFY THAT THE COUNTRY, DOCUMENT AND
C   THEME NUMBERS HAVE BEEN ACCURATELY TRANSCRIBED AND KEYPUNCHED
C

MAXC=149  $KDOC=0  $IFLAG=0  $NCCUN=0
READ(1,1)((NACCN(I,J),J=1,2),I=1,MAXC)
10 READ(6,2) ICCUN, IDOC, ITHEME, ILETER
   IF (EOF,6) 100,20
20 IF (ILETER.EQ.0) ILETER=NBLANK
   IF (IDOC.EQ.KDOC) GO TO 30
   KDOC=IDOC
   PRINT 4, IDOC
30 IF (ICCUN.EQ.NCCUN) PRINT 5, ITHEME, ILETER
   IF (ICCUN.NE.NCCUN) PRINT 5, (NACCN(ICCUN, J), J=1,2), ITHEME, ILETER
   NCCUN=ICCUN
   GO TO 10

C
C   FORMAT STATEMENTS
C
1 FORMAT(5X,A6,A6)
2 FORMAT(13,7X,13,6X,14,A1)
4 FORMAT(141,* THE FOLLOWING OBSERVATIONS ARE IN DOCUMENT*,15)
5 FORMAT(7,5X,A6,A6,4X,14,A1)
6 FORMAT(23X,14,A1)
100 END

```

PROGRAM REMOVE

Purpose

This program extracts all data on specified theme numbers from an active data tape. Its primary function is to delete all observations on outdated or inactive themes and punch them in a format that can be used to add them to the historical tape. Its secondary function is to equivalence two or more theme numbers with the same substantive meaning to a single number by deleting all observations on the "duplicate" theme numbers and punching each with the permanent number.

Method

For each theme that is to be retired, a flag is set in THEME which distinguishes a lettered from a non-lettered theme to be retired. If the retired theme is lettered, its number and letter are stored in NSPEC.

For each theme to be equivalenced, a flag is set in THEME which distinguishes lettered from non-lettered equivalences as well as from retired themes. If either the duplicate or the permanent theme contains a letter, then both are stored in NSPEC.

The active data tape is read a record at a time. If the record does not refer to a theme number being changed, it is written on the output tape. If it refers to a non-lettered theme being retired or equivalenced, the observation is deleted and punched. If, on the other hand, the record refers to a theme number which has been flagged as a lettered change, NSPEC is searched for this record's number and letter. If it is found, the record is deleted and punched. If it cannot be found, because this record's version is not being changed, the record is written on the output tape.

Dimensioned variables

There are four dimensioned variables in program REMOVE:

(1) THEME, dimensioned 1599, contains a field for each theme number. This array consists of a flag for each theme number to denote if and how it is being changed.

THEME (M):

- = O if theme M is non-existent or unchanged
- = M if all versions of theme M are being retired
- = -M if only lettered versions of theme M are being retired
- = K if all versions of theme M are being equivalenced to versions of theme K
- = -K if a version of theme M is being equivalenced to a version of theme K.
(Either M or K or both refer to lettered themes.)

(2) MTEMP, dimensioned 4, contains the temporary storage of the retired or equivalenced theme being processed. When retired themes are being stored, only the first two fields are filled with the theme number and letter, if any. When equivalences are being stored, the theme number and letter of the duplicate theme are followed by the theme number and letter of the permanent theme.

(3) NSPEC, dimensioned 500 by 4, contains the permanent storage of lettered themes being retired or equivalenced. It takes on the values of MTEMP whenever MTEMP (2) or MTEMP (4) contains a letter.

(4) RECORD, dimensioned 5, contains all the information from the record being processed, namely, country number, special codes, document number, theme number, and letter, if any.

Operational Effects of System Changes

Any changes in basic format of the input data tape could affect the way RECORD is read and how its fields are interpreted. Less dramatic effects are that if the maximum theme number exceeds 1599, MAX must be increased and THEME redimensioned; and if the number of retired or equivalenced themes involving a letter exceeds 500, NSPEC and NLETER must be increased.

Program Listing

The listing of program REMOVE follows.

```

PROGRAM REMOVE
DIMENSION THEME(1599),MTEMP(4),NSPEC(500,4),RECORD(5)
DATA(BLANK=1H )
TYPE INTEGER THEME,RECORD,BLANK

```

```

C
C INITIALIZE MAX NO OF THEMES, MAX NO OF RETIREES, MAX NO OF
C LETTERED CHANGES....LOGICAL UNIT OF INPUT AND OUTPUT
C TAPES, LOGICAL UNIT OF REMOVED RECORDS..ZERO OUT
C COUNTER AND ARRAY OF THEME FLAGS
C

```

```

MAX=1599 $NLETER=500
LIN=1 $LOUT=2 $LRET=62 $NCOUNT=0
DO 100 M=1,MAX

```

```

100 THEME(M)=0

```

```

C
C READ MAIN CONTROL CARD ...NUMBER OF RETIRED AND EQUIV THEMES
C

```

```

READ 1,NRET,NEQUIV

```

```

C
C SET UP FLAGS IN ARRAY THEME TO DENOTE RETIRED THEMES.....
C IF K IS A NON-LETTERED THEME THEME(K)=K AND IF NOT -K
C IF LETTERED ALSO FILL ARRAY NSPEC WITH K AND LETTER
C

```

```

DO 200 M=1,NRET
READ 2, (MTEMP(M),M=1,2)
PRINT 2, (MTEMP(M),M=1,2)
IF(MTEMP(M).EQ.BLANK) GO TO 150
NCOUNT=NCOUNT+1
IF(NCOUNT.GT.NLETER) GO TO 2000
NSPEC(NCOUNT,1)=MTEMP(1)
NSPEC(NCOUNT,2)=MTEMP(2)
NSPEC(NCOUNT,3)=0
THEME(MTEMP(1))=-1*MTEMP(1)
GO TO 200

```

```

150 THEME(MTEMP(1))=MTEMP(1)

```

```

200 CONTINUE

```

```

C
C READ EQUIV THEMES..IF REFER TO LETTERED THEMES FILL
C ARRAY NSPEC WITH INFO ON INPUT CARD AND SET THEME(K)
C EQUAL TO MINUS EQUIV OF K
C IF NON-LETTERED, ONLY SET THEME(K)=EQUIV THEME TO K
C

```

```

300 DO 400 I=1,NEQUIV
READ 2, (MTEMP(M),M=1,4)
PRINT 2, (MTEMP(M),M=1,4)
IF(MTEMP(2).EQ.BLANK.AND.MTEMP(4).EQ.BLANK) GO TO 350
NCOUNT=NCOUNT+1
IF(NCOUNT.GT.NLETER) GO TO 2000
DO 325 MM=1,4
325 NSPEC(NCOUNT,MM)=MTEMP(MM)

```

```

        THEME(MTEMP(1))=-1*MTEMP(3)
        GO TO 400
350  THEME(MTEMP(1))=MTEMP(3)
400  CONTINUE
C
C      READ A RECORD AND IF IT DOES NOT REFER TO A THEME
C      BEING CHANGED..WRITE ON THE OUTPUT TAPE
C
500  READ(LIN,3)(RECORD(M),M=1,5)
      IF(EOF,LIN)5000,600
600  MM=THEME(RECORD(4))
      IF(MM)800,650,700
650  WRITE(LOUT,3)(RECORD(M),M=1,5)
      GO TO 500
C
C      RECORD REFERS TO A NON-LETTERED THEME BEING CHANGED
C      PUNCH RECORD
C
700  IF(MM.EQ.RECORD(4)) WRITE(LRET,3)(RECORD(M),M=1,5)
      IF(MM.NE.RECORD(4)) WRITE(LRET,3)(RECORD(M),M=1,3),MM,BLANK
      GO TO 500
C
C      RECORD REFERS TO A LETTERED THEME BEING CHANGED
C      PUNCH RECORD IF IT CONTAINS THE LETTER BEING CHANGED
C      OTHERWISE WRITE ON OUTPUT TAPE
C
800  MM=IABS(MM)
      IFLAG=0
      DO 900 K=1,NCOUNT
        IF(RECORD(4).NE.NSPEC(K,1)) GO TO 900
        IF(RECORD(5).NE.NSPEC(K,2)) GO TO 900
        IF(NSPEC(K,3).EQ.0) WRITE(LRET,3)(RECORD(M),M=1,5)
        IF(NSPEC(K,3).NE.0) WRITE(LRET,3)(RECORD(M),M=1,3),NSPEC(K,3),NSPE
1C(K,4)
        IFLAG=1
900  CONTINUE
      IF(IFLAG.EQ.0) WRITE(LOUT,3)(RECORD(M),M=1,5)
      GO TO 500
C
C      PRINT ERROR MESSAGES CONCERNING OVERFLOWING DIMENSIONS
C
2000 PRINT 4,NLETER
      GO TO 500
C
C      ENDFILE OUTPUT TAPE
C
5000 ENDFILE LOUT
C
C      FORMAT STATEMENTS
C
1  FORMAT(2I5)
2  FORMAT(2(I4,A1))
3  FORMAT(I3,1X,I4,2X,I3,6X,I4,A1)
4  FORMAT(* TOO MANY LETTERED THEMES-EXCEEDS DIMENSION*,I5)
      END

```


PROGRAM UPDATE

Purpose

This program creates a new data tape by merging an existing one with an ordered card deck of additions and deletions. It preserves the basic structure of the data tape.

Method

A record and card are read and compared. The record is written on the output tape when any of the following conditions are met:

- (1) The record pertains to a smaller country number.
- (2) They both pertain to the same country
 - (a) and the record refers to a smaller theme number;
 - (b) and the same theme number, and the card is being added;
 - (c) and the same theme number, and the record is not identical with the delete card.
- (3) End of file has been encountered in the card deck.

The addition card is written on the output tape when any of the following conditions are met:

- (1) The card refers to a smaller country number than the record.
- (2) They are referring to the same country, and the card is for a smaller theme number.
- (3) End of file has been encountered on the input tape.

A deletion is accomplished whenever the card and record match on country, theme, and document number.

Dimensioned Variables

The data on the tape is read into RECORD, dimensioned 5. The update card is read into CARD, dimensioned 6. Both arrays are composed of the following information:

<u>Field</u>	<u>Description</u>
1	Country number
2	Remark codes <ol style="list-style-type: none">- regional code- repetition code- classification code- reliability code
3	Document number
4	Theme number
5	Letter associated with the theme, if any
6	Deletion (1) or addition (0 or blank) flag (CARD only)

Operational Effects of System Changes

If alterations are made to program UPDATE which change the structure or format of the data tape, corresponding alterations must be made to the format and possibly the logic of programs MATRIX, CHECK, REMOVE and LETTER.

Program Listing

The listing of program UPDATE follows.

```

PROGRAM UPDATE
DIMENSION RECORD(5),CARD(6)
TYPE INTEGER RECORD,CARD

```

```

C
C INITIALIZE EXISTING DATA TAPE TO LOGICAL UNIT 1,
C UPDATED OUTPUT TAPE TO LOGICAL UNIT 2 AND CARD
C DECK TO LOGICAL UNIT 60
C

```

```

LIN = 1 $ LOUT = 2 $ LCARD = 60

```

```

C
C READ A RECORD FROM THE EXISTING DATA TAPE
C

```

```

100 READ(LIN,1)(RECORD(M),M=1,5)
IF(EOF,LIN) 900,200

```

```

C
C READ A CARD FROM THE UPDATE DECK OF MODIFICATIONS
C

```

```

200 READ(LCARD,1)(CARD(M),M=1,6)
IF(EOF,LCARD) 800,300

```

```

C
C DOES CARD REFER TO SAME COUNTRY BEING PROCESSED
C ON THE TAPE
C

```

```

300 IF(RECORD(1)-CARD(1)) 400,500,600

```

```

C
C NO..RECORD REFERS TO EARLIER COUNTRY...
C WRITE THE RECORD AND READ ANOTHER RECORD
C

```

```

400 WRITE(LOUT,1)(RECORD(M),M=1,5)
READ(LIN,1)(RECORD(M),M=1,5)
IF(EOF,LIN) 900,300

```

```

C
C YES...THEY REFER TO SAME COUNTRY..
C DO THEY REFER TO THE SAME THEME....
C IF EARLIER THEME ON RECORD, WRITE RECORD
C IF EARLIER THEME ON CARD , WRITE CARD ON OUTPUT TAPE
C IF SAME THEME + ADDITION, WRITE THE RECORD
C IF SAME THEME + DELETION, COMPARE DOC + LETTER FIELDS
C IF MATCH, DELETE RECORD AND PRINT MESSAGE
C IF DO NOT MATCH, WRITE RECORD
C

```

```

500 IF(RECORD(4)-CARD(4)) 400,550,600
550 IF(CARD(6).EQ.0) GO TO 400
IF(RECORD(3).NE.CARD(3)) GO TO 400
IF(RECORD(5).NE.CARD(5)) GO TO 400
PRINT 2,(RECORD(M),M=1,5)
GO TO 100

```

```

C
C NO, CARD REFERS TO AN EARLIER COUNTRY
C IF ADDITION, WRITE CARD, READ ANOTHER CARD, PROCESS IT

```

```

C      IF DELETION, PRINT MESSAGE THAT NOT FOUND AND READ
C      THE NEXT CARD
C
600  IF (CARD(6).EQ.0) 650,700
650  WRITE(LOUT,1) (CARD(M),M=1,5)
      PRINT 3, (CARD(M),M=1,5), (RECORD(MM),MM=1,5)
      GO TO 200
700  PRINT 4, (CARD(M),M=1,5)
      GO TO 200

C
C      END OF FILE ENCOUNTERED ON THE CARD DECK,
C      CONTINUE TO READ/WRITE UNTIL EOF IS REACHED ON TAPE
C
800  WRITE(LOUT,1) (RECORD(M),M=1,5)
      READ(LIN,1) (RECORD(M),M=1,5)
      IF (EOF,LIN) 1000,800

C
C      END OF FILE ON THE TAPE, READ REMAINING CARDS
C      IF ADDITION, WRITE CARD ON TAPE
C      IF DELETION, PRINT ERROR MESSAGE THAT CARD WAS NOT FOUND
C
900  IF (CARD(6).EQ.0) WRITE(LOUT,1) (CARD(M),M=1,5)
      IF (CARD(6).NE.0) PRINT 4, (CARD(M),M=1,5)
      READ(LCARD,1) (CARD(M),M=1,6)
      IF (EOF,LCARD) 1000,900

C
C      END OF FILE ON CARD DECK AND TAPE...ENDFILE OUTPUT TAPE
C
1000 ENDFILE LOUT

C
C      FORMAT STATEMENTS
C
1  FORMAT(I3,1X,I4,2X,I3,6X,I4,A1,I1)
2  FORMAT(20X,4I5,A2,* WAS DELETED*)
3  FORMAT(* CARD*,4I5,A2,* WAS PLACED BEFORE RECORD*,4I5,A2)
4  FORMAT(4H****,* CARD*,4I5,A2,* CAN NOT BE FOUND ON THE TAPE*)
      END

```

MAINTAINING THE DICTIONARY TAPE

STRUCTURE OF THE DICTIONARY TAPE

The LOS dictionary tape consists of six files. Each program automatically accesses the file(s) it requires. The dictionary tape is written in BCD and is an unblocked, 7-track tape. The files are in the following order:

- (1) Country dictionary
- (2) Theme dictionary
- (3) Major/minor dictionary
- (4) Issue-variable dictionary
- (5) Lettered theme dictionary
- (6) Independent variable dictionary

For a detailed description of the format and structure of each of these files, see appendix B of the users guide.

PROGRAM SEARCH

Purpose

The primary function of program SEARCH is to locate specified themes within the major/minor and issue-variable dictionaries. It lists their assignments, but does not delete them. This program can be used to verify that every theme belongs to one and only one major and minor topic or to aid the user when retiring themes from the dictionary tape. It can process 1,000 themes in a single run.

Method

The program first reads the theme range selection cards. The array JCODE is filled with counters so that $JCODE(M) = N$ when theme M is the Nth theme to be processed. The program produces an overflow message if there are more than 1,000 themes to be processed.

Then the major/minor dictionary is read. If the Nth theme being processed is located in the major and minor topic, LOCATE(N, major topic) is set equal to the minor topic number. After all topics are read, LOCATE is printed whenever its cell is non-zero. A message is produced when all of its cells are zero (no assignment to a major and minor topic) or when more than one cell is non-zero (assigned to more than one major and minor topic).

Finally the issue-variable dictionary is read. If the Nth theme being processed is located within the issue-variable, LOCATE(N, counter) is set equal to the issue-variable number. The counter is increased as the cells are filled. The assignments to

issue-variables are printed in numerical order by theme number after the entire issue-variable dictionary has been read.

Dimensioned Variables

Program SEARCH contains four dimensioned variables:

(1) JCODE is dimensioned 1599, one field for each theme number. This array serves as a flag for the themes being processed. JCODE(M) = 0 if M is not being processed; JCODE(M) = N if M is the Nth theme processed.

(2) THEMES, dimensioned 500, contains the temporary storage of all themes in the minor topic or in the issue-variable being searched.

(3) LOCATE is dimensioned 1,000 by 20. Each row represents a theme being processed (up to a current maximum of 1,000). The columns take on different meanings for the major/minor and issue-variable dictionaries. When the assignments to topics are being stored, LOCATE(N, major topic) is zero when the Nth theme being processed is not assigned to this major topic and is equal to the minor topic when it is. When the assignments to issue-variables are being stored, LOCATE is filled across the row as the Nth theme is located. The Nth theme cannot belong to more than 20 issue-variables.

(4) KOUNT is an array dimensioned 1,000. Each field corresponds to a theme being processed. KOUNT is the counter used to fill LOCATE with assignments of themes to issue-variables. KOUNT(N) is increased by 1 everytime the Nth theme is located in the issue-variable dictionary.

Operational Effects of System Changes

Since this program is dependent upon the major/minor and issue-variable dictionaries, any change to their structure or format will have an operational effect. Numerous minor changes can be made easily. If the largest theme number in the theme dictionary is larger than 1599, NTHEME must be set equal to the new maximum and JCODE must be dimensioned at least as large. If the number of major topics is increased, NMAJ must be equal to the largest major topic and LOCATE must have at least as many columns as NMAJ. If the number of issue-variables is increased, NVAR must be equal to the largest issue-variable number. If themes are assigned to more than 20 issue-variables, LOCATE must be expanded. If more than 1,000 themes are to be processed in a single run, KOUNT and LOCATE must be expanded. If the number of themes assigned to any issue-variable or minor topic should ever exceed 500, THEMES must be expanded.

Program Listing

The listing of program SEARCH follows.

```

PROGRAM SEARCH
DIMENSION JCODE(1599),THEMES(500),KOUNT(1000),LOCATE(1000,20)
TYPE INTEGER THEMES,COUNT

```

```

C
C INITIALIZE NUMBER OF MAJOR TOPICS,NUMBER OF VARIABLES,MAXIMUM
C THEME NUMBER,LOGICAL UNIT OF DICTIONARY AND OUTPUT, COUNTER
C

```

```

NMAJ=17 $NVAR=70 $NTHEME=1599 $LDICT=1 $LOUT=61 $NCOUNT=0

```

```

C
C READ THEME RANGE SELECTION CARDS
C

```

```

100 READ 1,MIN,MAX
    IF(EOF,60)250,150
150 DO 200 M=MIN,MAX
    NCOUNT=NCOUNT+1
    IF(NCOUNT.GT.1000) GO TO 750
    KOUNT(NCOUNT)=0
200 JCODE(M)=NCOUNT
    GO TO 100

```

```

C
C READ MAJOR/MINOR DICTIONARY..FILL LOCATE WITH MAJ/MIN ASSIGNMENTS
C

```

```

250 DO 300 M=1,2
300 CALL SKIPFILE (LDICT)
    DO 450 MAJOR=1,NMAJ
    READ(LDICT,2)NMIN
    DO 400 MINOR=1,NMIN
    READ(LDICT,2)NO
    READ(LDICT,3)(THEMES(M),M=1,NO)
    DO 350 M=1,NO
    MM=JCODE(THEMES(M))
    IF(MM.EQ.0) GO TO 350
    IF(LOCATE(MM,MAJOR).NE.0) WRITE(LOUT,13)THEMES(M),NMIN,MAJOR
    LOCATE(MM,MAJOR)=MINOR
350 CONTINUE
400 CONTINUE
450 CONTINUE

```

```

C
C PRINT ASSIGNMENTS NUMERICALLY
C

```

```

PRINT 7
DO 550 INDEX=1,NTHEME
IFLAG=0
IF(JCODE(INDEX).EQ.0) GO TO 550
DO 500 MAJOR=1,NMAJ
MM=JCODE(INDEX)
IF(LOCATE(MM,MAJOR).EQ.0) GO TO 500
IF(IFLAG.EQ.1) WRITE(LOUT,6)INDEX
IF(IFLAG.EQ.0) IFLAG=1
WRITE(LOUT,4)INDEX,MAJOR,LOCATE(MM,MAJOR)

```

```

500 LOCATE(MM,MAJOR)=0
    IF(IFLAG.EQ.0) WRITE(LOUT,5) INDEX
550 CONTINUE
C
C   READ VARIABLE DICTIONARY
C
    CALL SKIPFILE (LDICT)
    PRINT 8
    DO 650 INDEX=1,NVAR
    READ(LDICT,2) NO
    READ(LDICT,3) (THEMES(M),M=1,NO)
    DO 600 M=1,NO
    MM=JCODE(THEMES(M))
    IF(MM.EQ.0) GO TO 600
    KOUNT(MM)=KOUNT(MM)+1
    IF(KOUNT(MM).GT.20) GO TO 800
    COUNT=KOUNT(MM)
    LOCATE(MM,COUNT)=INDEX
600 CONTINUE
650 CONTINUE
C
C   PRINT VARIABLE ASSIGNMENTS
C
    DO 700 INDEX=1,NTHEME
    IF(JCODE(INDEX).EQ.0) GO TO 700
    MM=JCODE(INDEX) $COUNT=KOUNT(MM)
    WRITE(LOUT,10) INDEX,(LOCATE(MM,M),M=1,COUNT)
700 CONTINUE
    GO TO 900
750 WRITE(LOUT,12)
    GO TO 900
800 WRITE(LOUT,11) INDEX
C
C   FORMAT STATEMENTS
C
1  FORMAT(2I5)
2  FORMAT(55X,I5)
3  FORMAT (20I4)
4  FORMAT(10X,I5,* IS IN MAJOR/MINOR*,2I3)
5  FORMAT(4H****,I5,* DOES NOT BELONG TO ANY TOPICS*)
6  FORMAT(4H****,I5,* BELONGS TO MORE THAN ONE TOPIC*)
7  FORMAT(1H1,* THE FOLLOWING ARE THE MAJOR/MINOR ASSIGNMENTS FOR TH
    1E FOLLOWING THEMES*,//)
8  FORMAT(1H1,* THE FOLLOWING ARE THE VARIABLE ASSIGNMENTS FOR THE F
    1OLLOWING THEMES*,//)
9  FORMAT(8(5X,I5))
10 FORMAT(10X,I5,* IS IN VARIABLES*,(10I5))
11 FORMAT(10X,I5,* IS IN MORE THAN 20 VARIABLES*)
12 FORMAT(* ERROR IN INPUT DECK...MORE THAN 1000 THEMES REQUESTED*)
13 FORMAT(4H****,* ADDITIONAL ERROR*,/* THEME*,I5,* IS ALSO ASSIGNED
    1 TO *,2I5)
900 END

```


PROGRAM DTAPE

Purpose

This program and its subroutines update the six dictionary files. Any file can be left unchanged or have specified entries changed. The first five files can also be completely updated with a new card deck.

Method

DTAPE consists of a main calling program and seven subroutines. The calling program reads the main control card, which specifies if and how a file is to be changed. The files are then processed one at a time. When the first five files are processed, the calling program copies the file on the output tape if it is to be left unchanged; reads and writes the new card deck on the tape if it is to be completely updated by cards; or calls the appropriate file subroutine (ONE through FIVE) if specific entries are to be changed. When file six is processed, subroutine SIX is always called.

Subroutine ONE performs all changes to specified entries in file 1, country dictionary. Additions and deletions of countries are not allowed, since they would affect the data on the raw data tape. However, the name of a country and its abbreviation can be corrected.

File 1 and the deck of changes are organized numerically by country number. A card and a record are read and compared. If the card refers to a smaller country number, it is out of order and subroutine ERROR is called. If they refer to the same country number, the name and abbreviation from the card are substituted for the record and written on the output tape. If the card refers to a larger country number, the record is written.

If a change card requests anything but a substitution or if an EOF is encountered on the input file before it has occurred on the change deck, subroutine ERROR is called. When an EOF is found on the change deck, all remaining input records are written on the output tape.

Subroutine TWO performs all changes to specified entries in file 2, theme dictionary. Additions, deletions and substitutions are allowed as long as the theme number is less than 1,600.

File 2 and the deck of changes are organized numerically by theme number. A card and a record are read and compared. If the card refers to a smaller theme number and requests an addition, the card is written on the output tape; if it does not request an addition, it is out of order and subroutine ERROR is called. If the card and record refer to the same theme number and the card requests an addition, subroutine ERROR is

called; if it requests a substitution, the theme title on the card is written; if it requests a deletion, the record is deleted. If the card refers to a larger theme number, the record is written on the output tape.

If a change card attempts to modify or create a theme larger than 1599 or if an EOF is encountered on the input file (blank record precedes its EOF) before it has occurred in the change deck, subroutine ERROR is called. When an EOF is found on the change deck, all remaining input records are written on the output tape.

Subroutine THREE performs all changes to specified entries in file 3, major/minor dictionary. The titles of the topics and the themes assigned to them can be changed. New major or minor topics cannot be created, nor can existing ones be deleted. File 3 is organized numerically by major topic number and, within each major topic, by minor topic number, each being followed by a list of its themes in numerical order. The deck of changes is similarly ordered.

A card and record are read and compared. If the card refers to a smaller major topic number, it is out of order and subroutine ERROR is called. If it refers to a larger major topic number, the major topic record is written on the output, and all of its minor topics and themes are read and written. If they refer to the same major topic, its title is changed, if requested, and written. Then a comparison is made of the minor topic numbers. If the card change refers to a smaller minor topic, subroutine ERROR is called; if it refers to a larger minor topic number, the minor topic and its themes are copied off the input file. If they refer to the same minor topic, its title is changed, if requested. The change cards are then compared with the list of themes in the minor topic; the additions are made in the appropriate numerical location and deletions are made when they are located in the list. If deletions cannot be found or entries are out of numerical order, subroutine ERROR is called. Once all changes have been made, the minor topic title and the number of themes now assigned to it are written, followed by a list of these theme numbers.

If an EOF is encountered on the input file before it occurs on the change deck, subroutine ERROR is called. When all changes have been made, all remaining records are written on the output tape.

Subroutine FOUR performs all changes to specified entries in file 4, issue-variable dictionary. The titles of the issue-variables and the ranks and themes within them can be changed. Existing issue-variables can be deleted, which creates an unused number, and new issue-variables can be created for any of the unused numbers between 1 and 70.

File 4 is organized numerically by issue-variable number and, within each issue-variable, by rank of themes. The deck of changes is similarly ordered. This subroutine reads an issue-variable identification card and record. If the card refers to a smaller issue-variable number, it is out of order and subroutine ERROR is called. If

it refers to a larger issue-variable number, the record is written, all of its ranks and themes are read and written, and the next issue-variable identification record is read.

If the card and record refer to the same issue-variable, its title is changed, if requested. If only the title is being changed, the new title is written, and the existing ranks and themes are read and written.

If there are rank and theme changes to be made, the existing ones and changes are read. Then one at a time, a record and card entry are compared. The card additions are made in the appropriate rank order and deletions are made when they are located on the list. If deletions cannot be found or if a card entry is out of rank order, subroutine ERROR is called. Once all changes have been made, the issue-variable title and the number of themes now assigned to it are written, followed by a list of the ranks and themes.

If an EOF is encountered on the change deck, all remaining records on the file are copied onto the output tape. If an EOF occurs on the input file before it is encountered on the change deck, subroutine ERROR is called.

Subroutine FIVE performs all changes to specified entries in file 5, lettered theme dictionary. Additions, deletions and substitutions of lettered themes are processed by this subroutine. File 5 and the deck of changes are organized numerically by theme number and, within each theme number, alphabetically by letter.

A card and record are read and compared. If the card refers to a smaller theme number and is an addition, it is written on the output tape and another card is read; but if it is not an addition, it is out of order and subroutine ERROR is called. If the card refers to a larger theme number, the record is written on the output tape and the next record is read. If the card and record refer to the same theme, their letters are compared. If they do not match, the record is written and the next one processed. But if they do match and it is an addition, subroutine ERROR is called; if it is a deletion, the record is deleted and the next record and card are compared; if it is a substitution, the card title is substituted and written on the output tape, and the next record and card are compared.

When an EOF has been encountered on the change deck, all remaining records are written on the output tape. If an EOF is encountered on the input tape and changes still remain, the additions are placed on the output tape, but any deletions or substitutions call subroutine ERROR.

Subroutine SIX performs all operations on file 6, independent variable dictionary. If the file is to be left unchanged, it is copied. If a complete update is requested, a message is produced that the request is unacceptable and the file is copied unchanged. If specific entries are to be changed, a country's data is read off the input and compared with a card in the change deck. If the card refers to a smaller country number, an error message

is written and the entire file is left unchanged. If the card refers to a larger country number, the record is written on the output tape and the next country's data is read. If they refer to the same country the card entry is substituted for the record entry and another card is read.

If an EOF occurs on the change deck, all remaining records are copied onto the output tape. If an EOF is encountered on the input file before it is found on the change deck, an error message is produced, the changes are disregarded, and the entire file is copied unchanged.

Subroutine ERROR is called whenever a deck of changes is out of order or is incompatible with the above subroutines. This subroutine erases all work on a file, rewinds the input and output tapes and copies the file unchanged onto the output tape. Any remaining changes to this file are skipped over.

Structure of the Output Tape

The six files that are on the output tape have the same format and structure as the input tape. (For a detailed description of the dictionary file structure, see the users guide.)

Dimensioned Variables

There are five dimensioned variables that appear in the calling program and the seven subroutines. Most of them take on different meanings depending on the subroutine. Their general meaning will be discussed here.

(1) NFILE, dimensioned 6, contains a change code for each file. If NFILE(M) is zero, file M is to be left unchanged; if it is 1, file M is to be completely updated with cards; if it is 2, specified entries are to be changed.

(2) INFO, dimensioned 500, contains all of the information from a record. The reader should refer to the users guide for a description of a record in each file.

(3) NINFO, dimensioned 500, contains all of the information from a change card. The reader should refer to the users guide for a description of the format and structure of the change cards for each file.

(4) VDATA, dimensioned 500 by 2, is used as temporary storage of the themes assigned to the minor topic being processed in subroutine THREE (only the first column is used) or the temporary storage of the ranks and themes assigned to the issue-variable being processed in subroutine FOUR.

(5) NVDATA, dimensioned 50 by 3, is used as temporary storage of changes to themes in the minor topic or issue-variable being changed. In subroutine THREE, the first two columns of NVDATA are filled with a + (addition) or - (deletion), followed by the theme number. In subroutine FOUR, the three columns are used to store the rank, + or -, and theme change.

Operational Effects of System Changes

If the format or structure of the dictionary tape is changed, all the programs in the system, especially this one, will have to be changed. DTAPE is not affected by changes in any other program, but all other programs depend upon its product, namely the dictionary files. Appendix A identifies how changes can affect each of the programs in the system.

Program listing

The program and subroutine listings follow.

```

PROGRAM DTAFE
COMMON NFILE(6),INFC(500),NINFO(500),VDATA(500,2),NVDATA(50,3)
DATA(PLUS=1H+),(MINUS=1H-),(BLANK=1H )
TYPE INTEGER VDATA
C
C LOGICAL UNIT OF THE INPUT AND OUTPUT ARE SET..NUMBER OF MAJOR TOPICS
C AND VARIABLES....READ WHICH FILES ARE TO BE CHANGED
C
C CHECK ONE FILE AT A TIME TO SEE HOW CHANGING IT
C IF NOT CHANGING THIS FILE, COPY FILE ONTO OUTPUT TAPE
C IF COMPLETELY UPDATING THIS FILE WITH CARDS, WRITE CARD DECK
C ON THE OUTPUT TAPE
C
      LIN=1 $ LOUT=2 $INDEX=1 $NMAJ=17 $NVAR=70 $NDISC=3
      READ 1,(NFILE(M),M=1,6)
100 IF(NFILE(INDEX)-1) 150,200,300
150 LU=1
      PRINT 3,INDEX
      GO TO 250
200 CALL SKIPFILE (LIN)
      LU=60
      PRINT 4,INDEX
250 READ(LU,2)(INFO(M),M=1,10)
      IF(EOF,LU) 500,275
275 WRITE(LOUT,2)(INFO(M),M=1,10)
      IF(LU.EQ.60) PRINT 5,(INFC(M),M=1,10)
      GO TO 250
500 ENDFILE LOUT
      INDEX=INDEX+1
      IF(INDEX.LE.5) 100,6000
C
C IF EDITING THE FILE GO TO THE APPROPRIATE SUBROUTINE
C OTHERWISE COPY FILE OR CARD DECK ONTO OUTPUT TAPE
C
300 GO TO (1000,2000,3000,4000,5000),INDEX
1000 CALL ONE
      INDEX=2
      GO TO 100
2000 CALL TWO
      INDEX=3
      GO TO 100
3000 CALL THREE
      INDEX=4
      GO TO 100
4000 CALL FOUR
      INDEX=5
      GO TO 100
5000 CALL FIVE
      INDEX=6
6000 CALL SIX
C
C FORMAT STATEMENTS
C
1 FORMAT(5I1)
2 FORMAT(10A8)
3 FORMAT(//,* FILE*,I5,* WAS NOT CHANGED*)
4 FORMAT(//,* FILE*,I5,* WAS COMPLETELY CHANGED WITH THESE CARDS*)
5 FORMAT(1X,10A4)
      END

```

```

SUBROUTINE ONE
COMMON NFILE(6),INFC(500),NINFO(500),VDATA(500,2),NVDATA(50,3)
DATA(PLUS=1H+),(MINUS=1H-),(BLANK=1H )
TYPE INTEGER VDATA
LIN=1 $ LOUT=2 $INDEX=1 $NMAJ=17 $NVAR=70 $NOISC=3
1 FORMAT(//,* THE FOLLOWING CHANGES WERE MADE TO FILE*,I5)
3 FORMAT(I3,2X,A8,A6,A3,57X,A1)
6 FORMAT(* SUBSTITUTION...*,I5,A8,A6,A3,* WAS ORIGINALLY*,I5,A8,A6,A
13)

```

```

C-----
C   THIS SUBROUTINE MAKES SPECIFIED CHANGES TO FILE 1...
C   SUBSTITUTIONS ARE ALLOWED..ADDITIONS AND DELETIONS ARE NOT
C   THEY PRODUCE AN ERROR AND THE FILE IS LEFT UNCHANGED
C-----

```

```

PRINT 1,INDEX
1000 READ(60,3) (NINFO(M),M=1,4),SIGN
IF(EOF,60) 1900,1100
1100 IF(SIGN.EQ.BLANK) GO TO 1200
1150 CALL ERROR (INDEX)
GO TO 2000
1200 READ(LIN,3) (INFO(M),M=1,4)
IF(EOF,LIN) 1150,1300

```

```

C
C   COMPARE A CARD AND RECORD
C
1300 IF(NINFO(1)-INFO(1)) 1150,1500,1600

```

```

C
C   SUBSTITUTION
C
1500 PRINT 6, (NINFO(M),M=1,4), (INFO(MM),MM=1,4)
WRITE(LOUT,3) (NINFO(M),M=1,4),INDEX
GO TO 1000

```

```

C
C   CARD HAS LARGER COUNTRY NUMBER...WRITE RECORD
C
1600 WRITE(LOUT,3) (INFO(M),M=1,4),INDEX
GO TO 1200

```

```

C
C   EOF ON CHANGE DECK..WRITE REMAINING FILE ENTRIES
C

```

```

1900 READ(LIN,3) (INFO(M),M=1,4)
IF(EOF,LIN) 1999,1950
1950 WRITE(LOUT,3) (INFO(M),M=1,4),INDEX
GO TO 1900
1999 ENDFILE LOUT
2000 RETURN
END

```

```

SUBROUTINE TWO
COMMON NFILE(6),INFO(500),NINFO(500),VDATA(500,2),NVDATA(50,3)
DATA(PLUS=1H+),(MINUS=1H-),(BLANK=1H )
TYPE INTEGER VDATA
LIN=1 $ LOUT=2 $ INDEX=2 $ NMAJ=17 $ NVAR=70 $ NDISC=3
1 FORMAT(//,* THE FOLLOWING CHANGES WERE MADE TO FILE*,I5)
6 FORMAT(* SUBSTITUTION...*,I5,6A8,* WAS ORIGINALLY*,I5,4A8)
7 FORMAT(I5,6A8,26X,A1)
10 FORMAT(* ADDED*,I5,6A8,* BEFORE RECORD *,I5,4A8)
12 FORMAT(* DELETED*,I5,6A8)
C-----
C   THIS SUBROUTINE MAKES SPECIFIED CHANGES TO FILE 2...
C   ONE CAN SUBSTITUTE, ADD OR DELETE THEMES AS LONG AS THEY
C   ARE NUMBERED LESS THAN 1600
C-----
      PRINT 1,INDEX
2000 READ(60,7)(NINFO(M),M=1,7) ,SIGN
      IF(EOF,60)2950,2100
2100 IF(NINFO(1).LE.1599)2200,2325
2200 READ(LIN,7)(INFO(M),M=1,7)
2250 IF(INFO(1).EQ.0) GO TO 2325
C
C   COMPARE A CARD AND RECORD
C
      IF(NINFO(1)-INFO(1))2300,2400,2500
C
C   CARD FOR SMALLER THEME NO..ERROR IF NOT ADD,OTHERWISE ADD CARD
C
2300 IF(SIGN.EQ.PLUS) GO TO 2350
2325 CALL ERROR(INDEX)
      GO TO 3000
2350 PRINT 10,(NINFO(M),M=1,7),(INFO(MM),MM=1,5)
      WRITE(LOUT,7)(NINFO(M),M=1,7),INDEX
      READ(60,7)(NINFO(M),M=1,7),SIGN
      IF(EOF,60)2900,2375
2375 IF(NINFO(1).LE.1599)2250,2325
C
C   IF SAME THEME NO THEN...
C       IF ADDITION, ERROR
C       IF DELETION, DELETE
C       IF BLANK, SUBSTITUTE
C
2400 IF(SIGN.EQ.PLUS) GO TO 2325
2450 IF(SIGN.EQ.BLANK)2455,2475
2455 PRINT 6,(NINFO(M),M=1,7),(INFO(MM),MM=1,5)
      WRITE(LOUT,7)(NINFO(M),M=1,7),INDEX
      GO TO 2000
2475 PRINT 12,(NINFO(M),M=1,7)
      GO TO 2000
C

```



```

C      CARD REFERS TO LARGER THEME NO...WRITE RECORD
C
2500  WRITE(LOUT,7)(INFO(M),M=1,7),INDEX
      GO TO 2200
C
C      EOF ON CHANGE DECK, WRITE REST OF INPUT FILE
C
2900  WRITE(LOUT,7)(INFO(M),M=1,7),INDEX
2950  READ( LIN ,7)(INFO(M),M=1,7)
      IF(INFO(1).EQ.0) 2999,2900
2999  WRITE(LOUT,7)(INFO(M),M=1,7),INDEX
      CALL SKIPFILE (LIN)
      ENDFILE  LOUT
3000  RETURN
      END

```

```

SUBROUTINE THREE
COMMON NFILE(5),INFO(500),NINFO(500),VDATA(500,2),NVDATA(50,3)
DATA(PLUS=1H+),(MINUS=1H-),(BLANK=1H )
TYPE INTEGER VDATA
TYPE INTEGER PLUS
LIN=1 $ LOUT=2 $ INDEX=3 $ NMAJ=17 $ NVAR=70 $ NDISC=3
1 FORMAT(//,* THE FOLLOWING CHANGES WERE MADE TO FILE*,I5)
2 FORMAT(10A8)
13 FORMAT(I5,I5,I1,I4,6A8)
14 FORMAT(I5,6A8,4X,I3,19X,I1)
18 FORMAT(20I4)
23 FORMAT(* THEME*,I5,* ADDED TO *,2I5)
27 FORMAT(16(A1,I4))
28 FORMAT(* THEME*,I5,* WAS DELETED FROM MAJOR/MINOR*,2I5)
30 FORMAT(I5)
55 FORMAT(* TITLE CHANGED FOR MAJOR*,I5,* TO*,6A8)
56 FORMAT(* TITLE CHANGED FOR MINOR*,I5,* TO*,6A8)
C-----
C THIS SUBROUTINE MAKES SPECIFIED CHANGES TO FILE 3...
C TITLES OF MAJOR AND MINOR TOPICS CAN BE CHANGED AND THEMES ASSIGNED
C TO THEM CAN BE CHANGED.. NEW TOPICS CAN NOT BE CREATED
C OR OLD TOPICS DELETED
C-----
PRINT 1,INDEX
3000 NCOUNT=0
READ(LIN,14)(INFO(M),M=1,7),NMIN
IF(EOF,LIN) 3900,3050
3050 READ(60,13)(NINFO(M),M=1,10)
IF(EOF,60)3950,3100
3100 IF(NINFO(1)-INFO(1))3200,3400,3250
3200 CALL ERROR(INDEX)
GO TO 4000
3250 WRITE(LOUT,14)(INFO(M),M=1,7),NMIN,INDEX
KK=1
3255 DO 3275 K=KK,NMIN
READ(LIN,14)(INFO(M),M=1,7),NTHEME
WRITE(LOUT,14)(INFO(M),M=1,7),NTHEME,INDEX
READ(LIN,18)(VDATA(M,1),M=1,NTHEME)
3275 WRITE(LOUT,18)(VDATA(M,1),M=1,NTHEME)
READ(LIN,14)(INFO(M),M=1,7),NMIN
IF(EOF,LIN)3200,3100
C
C SAME MAJOR TOPIC..TITLE CHANGE
C
3400 IF(NINFO(2).LE.0) 3425,3450
3425 DO 3430 M=5,10
MM=M-3
3430 INFO(MM)=NINFO(M)
PRINT 55,(INFO(K),K=1,7)
GO TO 3050

```

```

C
C SAME MAJOR BUT NO TITLE CHANGE
C
3450 NVAR=INFO(1)
      NCOUNT=NCOUNT+1
      WRITE(LOUT,14) (INFO(M),M=1,7),NMIN,INDEX
3455 READ(LIN,14) (INFO(M),M=1,7),NTHEME
C
C COMPARE MINOR TOPICS
C
3458 IF(NINFO(2)-INFO(1))3200,3500,3600
C
C SAME MINOR, CHANGE TITLE IF REQUESTED..READ EXISTING THEMES IN MINOR
C
3500 REWIND NDISC
      IF(NINFO(3).NE.1) GO TO 3505
      DO 3501 M=5,10
      MM=M-3
3501 INFO(MM)=NINFO(M)
      PRINT 56,(INFO(M),M=1,7)
3505 MTHEME=NTHEME
      READ(LIN,18) (VDATA(M,1),M=1,NTHEME)
      MM=NINFO(4)
C
C IF CORRECTIONS ARE TO BE MADE READ THEM...
C IF NOT WRITE THIS MINOR ON THE OUTPUT TAPE AND PROCESS NEXT CARD
C AND RECORD
C
      IF(MM.EQ.0) 3510,3520
3510 WRITE(LOUT,14) (INFO(M),M=1,7),NTHEME,INDEX
      WRITE(LOUT,18) (VDATA(M,1),M=1,NTHEME)
      GO TO 3548
C
C MAKE APPROPRIATE CORRECTIONS TO THIS MINOR TOPIC
C
3520 READ(60,27) ((NVDATA(M,N),N=1,2),M=1,MM)
      MCOUNT=1
      DO 3545 M=1,MM
3525 IF(MCOUNT.GT.NTHEME) GO TO 3530
      IF(NVDATA(M,2)-VDATA(MCOUNT,1))3530,3535,3540
3530 IF(NVDATA(M,1).EQ.PLUS) 3531,3200
3531 WRITE(NDISC,30) NVDATA(M,2)
      PRINT 23,NVDATA(M,2),NINFO(1),INFO(1)
      MTHEME=MTHEME+1
      GO TO 3545
3535 IF(NVDATA(M,1).EQ.PLUS) GO TO 3200
      PRINT 28,VDATA(MCOUNT,1),NVAR,INFO(1)
      MCOUNT=MCOUNT+1 $MTHEME=MTHEME-1 $ GO TO 3545
3540 WRITE(NDISC,30) VDATA(MCOUNT,1)
      MCOUNT=MCOUNT+1 $GO TO 3525

```

```

3545 CONTINUE
C
C      WRITE REMAINING CORRECTIONS AFTER CORRECTIONS HAVE BEEN MADE..
C      WRITE MINOR TOPIC TITLE AND THE NEW LIST OF THEMES..PROCESS NEXT MINOR
C
      IF (MCCOUNT.LT.NTHEME) 3546,3549
3546 DO 3547 K=MCCOUNT,NTHEME
3547 WRITE (NDISC,30) VDATA(K,1)
3549 ENDOFILE NDISC
      REWIND NDISC
      WRITE (LOUT,14) (INFO(K),K=1,7),MTHEME,INDEX
      READ (NDISC,30) (VDATA(K,1),K=1,MTHEME)
      WRITE (LOUT,18) (VDATA(M,1),M=1,MTHEME)
3548 NCOUNT=NCOUNT+1
      IF (NCOUNT.GE.NMIN) 3000,3550
3550 READ (60,13) (NINFO(M),M=1,10)
      IF (EOF,60) 3955,3555
3555 READ (LIN,14) (INFO(M),M=1,7),NTHEME
      IF (EOF,LIN) 3200,3556
3556 IF (NINFO(1)-NVAR) 3200,3458,3600
C
C      CARD REFERS TO LATER MINOR TOPIC...WRITE THIS MINOR AS IS
C
3600 WRITE (LOUT,14) (INFO(M),M=1,7),NTHEME,INDEX
      READ (LIN,18) (VDATA(M,1),M=1,NTHEME)
      WRITE (LOUT,18) (VDATA(M,1),M=1,NTHEME)
      NCOUNT=NCOUNT+1
      IF (NCOUNT.LE.NMIN) 3555,3557
3557 NCOUNT=0
      READ (LIN,14) (INFO(M),M=1,7),NMIN
      IF (EOF,LIN) 3200,3100
C
C      EOF ON CHANGE DECK..WRITE REMAINING INPUT ENTRIES
C
3949 WRITE (LOUT,14) (INFO(M),M=1,7),NTHEME,INDEX
      GO TO 3955
3950 WRITE (LOUT,14) (INFO(M),M=1,7),NMIN,INDEX
3955 READ (LIN,2) (INFO(M),M=1,10)
      IF (EOF,LIN) 3910,3960
3960 WRITE (LOUT,2) (INFO(M),M=1,10)
      GO TO 3955
3900 READ (60,13) (NINFO(M),M=1,10)
      IF (EOF,60) 3910,3200
3910 ENDOFILE LOUT
4000 RETURN
      END

```

SUBROUTINE FOUR

COMMON NFILE(6),INFO(500),NINFO(500),VDATA(500,2),NVDATA(50,3)
DATA(PLUS=1H+),(MINUS=1H-),(BLANK=1H)
TYPE INTEGER VDATA,PLUS

 LIN=1 \$ LOUT=2 \$INDEX=4 \$NMAJ=17 \$NVAR=70 \$NOISC=3
 1 FORMAT(//,* THE FOLLOWING CHANGES WERE MADE TO FILE*,I5)
 2 FORMAT(* VARIABLE*,I5,* WAS GIVEN A NEW RANK/THEME*,2I5)
 3 FORMAT(* VARIABLE*,I5,* WAS GIVEN A NEW TITLE*,6A8)
 14 FORMAT(I5,6A8,4X,I3,19X,I1)
 15 FORMAT(20I4)
 16 FORMAT(I5,I1,I3,6A8)
 17 FORMAT (8(I5,A1,I4))
 18 FORMAT(8(2I5))
 20 FORMAT(* VARIABLE*,I5,* RANK/THEME WAS DELETED*,2I5)
 21 FORMAT(2I5)

C-----

C THIS SUBROUTINE MAKES SPECIFIED CHANGES TO FILE 4.....

C-----

 PRINT 1,INDEX
4000 READ(60,16)(NINFO(M),M=1,9)
 IF(EOF,60)4997,4050
4050 READ(LIN,14)(INFO(M),M=1,7),NTHEME
 IF(EOF,LIN) 4200,4100

C

C COMPARE A CARD AND RECORD FOR VARIABLE NUMBER

C

4100 IF(NINFO(1)-INFO(1)) 4200,4400,4300

C

C CARD IS OUT OF ORDER

C

4200 CALL ERROR(INDEX)
 GO TO 5000

C

C CARD REFERS TO LARGER VAR NO..COPY ENTIRE VARIABLE ONTO OUTPUT TAPE

C

4300 WRITE(LOUT,14)(INFO(M),M=1,7),NTHEME,INDEX
 READ (LIN ,18)((VDATA(M,N),N=1,2),M=1,NTHEME)
 WRITE(LCUT,18)((VDATA(M,N),N=1,2),M=1,NTHEME)
 GO TO 4050

C

C SAME VARIABLE..CHANGE TITLE IF REQUESTED

C

4400 IF(NINFO(2).EQ.1) 4450,4500
4450 DO 4475 M=2,7
 MM=M+2
4475 INFO(M)=NINFO(MM)
 PRINT 3,(INFO(M),M=1,7)
4500 MM=NINFO(3)

```

C
C      IF NO CHANGES ARE TO BE MADE, WRITE EXISTING VAR ON OUTPUT TAPE
C      OTHERWISE READ EXISTING RANKS/THEMES AND CHANGES
C
      IF (MM .EQ. 0)      4510,4525
4510 READ(60,16) (NINFO(M),M=1,9)

      IF (EOF,60) 4995,4300
4525 READ(LIN,18) ((VDATA(M,N),N=1,2),M=1,NTHEME)
      READ(60,17) ((NVDATA(M,N),N=1,3),M=1,MM)
      MTHEME=NTHEME $REWIND NDISC $MM=1

C
C      PLACE ENTRY ADDITIONS IN APPROPRIATE RANK LOCATIONS OR DELETE THEME
C      WHEN ENTRIES MATCH
C
      DO 4800 M=1,NTHEME
4600 IF (NVDATA(MM,1)-VDATA(M,1)) 4650,4700,4750
4650 IF (NVDATA(MM,2).EQ.PLUS) 4660,4200
4660 WRITE(NDISC,21) NVDATA(MM,1),NVDATA(MM,3)
      PRINT 2,NINFO(1),NVDATA(MM,1),NVDATA(MM,3)
      MTHEME=MTHEME+1
4670 MM=MM+1
      IF (MM.GT.NINFO(3)) 4850,4600
4700 IF (NVDATA(MM,2).EQ.PLUS) GO TO 4725
      IF (NVDATA(MM,3).EQ.VDATA(M,2)) 4715,4750
4715 PRINT 20,NINFO(1),NVDATA(MM,1),NVDATA(MM,3)
      MTHEME=MTHEME-1 $MM=MM+1
      IF (MM.GT.NINFO(3)) 4849,4800
4725 IF (NVDATA(MM,3)-VDATA(M,2)) 4660,4200,4750
4750 WRITE(NDISC,21) (VDATA(M,N),N=1,2)
4800 CONTINUE
4805 IF (MM.LE.NINFO(3)) 4810,4900
4810 IF (NVDATA(MM,1).GE.VDATA(MTHEME,1)) 4815,4200
4815 IF (NVDATA(MM,2).NE.PLUS) GO TO 4200
      WRITE(NDISC,21) NVDATA(MM,1),NVDATA(MM,3)
      PRINT 2,NINFO(1),NVDATA(MM,1),NVDATA(MM,3)
      VDATA(MTHEME,1)=NVDATA(MM,1)
      MM=MM+1 $MTHEME=MTHEME+1 $GO TO 4805
4849 M=M+1
4850 DO 4875 KX=M,NTHEME
4875 WRITE(NDISC,21) (VDATA(KX,N),N=1,2)

C
C      WRITE NEWLY CREATED VARIABLE ON OUTPUT
C
4900 ENDFILE NDISC
      REWIND NDISC
      WRITE(LOUT,14) (INFO(M),M=1,7),MTHEME,INDEX
      READ(NDISC,21) ((VDATA(L,N),N=1,2),L=1,MTHEME)
      WRITE(LOUT,18) ((VDATA(L,N),N=1,2),L=1,MTHEME)
      GO TO 4000

```

```

C
C      EOF ON CHANGE DECK, WRITE REMAINING FILE ENTRIES ON OUTPUT
C
4998 WRITE(LOUT,14) (INFO(M),M=1,7),NTHEME,INDEX
      READ(LIN,18) ((VDATA(M,N),N=1,2),M=1,NTHEME)
      WRITE(LOUT,18) ((VDATA(M,N),N=1,2),M=1,NTHEME)
4997 READ(LIN,14) (INFO(M),M=1,7),NTHEME,INDEX

      IF(EOF,LIN)4999,4998
4999 ENDFILE LOUT
5000 RETURN
      END

```

```

SUBROUTINE FIVE
COMMON NFILE(6),INFO(500),NINFO(500),VDATA(500,2),NVDATA(50,3)
DATA(PLUS=1H+),(MINUS=1H-),(BLANK=1H )
TYPE INTEGER VDATA
TYPE REAL MINUS
LIN=1 $ LOUT=2 $INDEX=5 $NMAJ=17 $NVAR=70 $NDISC=3
1 FORMAT(//,* THE FOLLOWING CHANGES WERE MADE TO FILE*,I5)
6 FFORMAT(* SUBSTITUTION*,I5,A1,6A8,* WAS ORIGINALLY*,I5,A1,3A8)
10 FFORMAT(* ADDED*,I5,A1,6A8,* BEFORE RECORD*,I5,A1,3A8)
12 FFORMAT(* DELETED*,I5,A1,6A8)
22 FFORMAT(I5,A1,6A8,25X,A1)
C-----
C THIS SUBROUTINE MAKES SPECIFIED CHANGES TO FILE 5.....
C SUBSTITUTES,ADDS AND DELETES THEMES WITH LETTERS
C-----
PRINT 1,INDEX
5000 READ(60,22)(NINFO(M),M=1,8),SIGN
IF(EOF,60)5950,5100
5100 READ(LIN,22)(INFO(M),M=1,8)
IF(EOF,LIN) 5998,5200
C
C COMPARE A CARD AND RECORD
C
5200 IF(NINFO(1)-INFO(1))5300,5400,5500
C
C CARD REFERS TO SMALLER THEME NO. ...ADD IF ADDITION, OTHERWISE ERROR
C
5300 IF(SIGN.EQ.PLUS)5350,5460
5350 PRINT 10, (NINFO(M),M=1,3),(INFO(MM),MM=1,5)
WRITE(LOUT,22)(NINFO(M),M=1,8),INDEX
GO TO 5600
C
C SAME THEME NO. BUT DIFFERENT LETTERS..RECORD IS WRITTEN
C SAME LETTER AS WELL, PERFORM SUBSTITUTION OR DELETION
C
5400 IF(NINFO(2).EQ.INFO(2)) 5450,5500
5450 IF(SIGN.EQ.PLUS)5460,5470
5460 CALL ERROR(INDEX)
GO TO 6000
5470 IF(SIGN.EQ.MINUS) PRINT 12,(NINFO(M),M=1,8)
IF(SIGN.EQ.MINUS) GO TO 5300
IF(SIGN.EQ.BLANK) PRINT 6,(NINFO(M),M=1,8),(INFO(MM),MM=1,5)
WRITE(LOUT,22)(NINFO(M),M=1,8),INDEX
GO TO 5000
C
C CARD REFERS TO LARGER THEME NO...WRITE RECORD
C
5500 WRITE(LOUT,22)(INFO(M),M=1,8),INDEX
GO TO 5100
5600 READ(60,22)(NINFO(M),M=1,8),SIGN

```



```

      IF(EOF,60)5900,5200
C
C      EOF ON CHANGE DECK, COPY REMAINING RECORDS
C
5900 WRITE(LOUT,22)(INFO(M),M=1,8),INDEX
5950 READ(LIN,22 )(INFO(M),M=1,8)
      IF(EOF,LIN) 5999,5900
C
C      EOF ON INPUT TAPE, COPY REMAINING ADDITIONS FROM CARD DECK
C      IF DECK INCLUDES DELETIONS OR SUBSTITUTIONS, ERROR IS CALLED
C
5998 IF(SIGN.EQ.PLUS) WRITE(LOUT,22)(NINFO(M),M=1,8),INDEX
      IF(SIGN.NE.PLUS) GO TO 5460
      READ(60,22)(NINFO(M) ,M=1,8),SIGN
      IF(EOF,60)5999,5998
5999 ENDFILE LOUT
6000 RETURN
      END

```

```

SUBROUTINE SIX
COMMON NFILE(6),INFO(500),NINFO(500),VDATA(500,2),NVDATA(50,3)
TYPE INTEGER VDATA
TYPE REAL NINFO,INFO,NO
DATA(PLUS=1H+),(MINUS=1H-),(BLANK=1H )
LIN=1 $ LOUT=2 $ INDEX=6 $ NMAJ=17 $ NVAR=70 $ NDISC=3
1 FORMAT(/,*,* THE FOLLOWING CHANGES WERE MADE TO FILE*,I5)
2 FORMAT(* COUNTRY*,F5.0,* ENTRY*,F5.0,* WAS CHANGED TO* F10.3,* WAS
1*,F10.3)
3 FORMAT(/,*,* FILE*,I5,* WAS NOT CHANGED*)
23 FORMAT(* FILE*,I5,*HAD A CARD OUT OF ORDER AND THE FILE WAS LEFT U
INCHANGED...*,/,* ....DISREGARD CORRECTIONS ABOVE*)
24 FORMAT(* FILE 6 CAN NOT BE UPDATED COMPLETELY WITH CARDS*)
25 FORMAT(F4.0,34F3.1,/, (12F10.1))
26 FORMAT(2F5.0,F20.6)
C-----
C THIS SUBROUTINE MAKES ALL CHANGES TO FILE 6...
C ONE CAN LEAVE THIS FILE UNCHANGED OR MAKE SPECIFIED CHANGES
C ONE CAN NOT COMPLETELY UPDATE THIS FILE WITH CARDS
C-----
6000 IF(NFILE(6)-1)6199,6150,6299
C
C PRINT MESSAGE THAT FILE 6 CANT BE COMPLETELY UPDATED WITH CARDS
C
6150 PRINT 24
C
C NO CHANGES TO THIS FILE...COPY FILE
C
6199 PRINT 3,INDEX
6200 READ(LIN,25) (INFO(M),M=1,104)
IF(EOF,LIN)7000,6250
6250 WRITE(LOUT,25) (INFO(M),M=1,104)
GO TO 6200
C
C PERFORM SPECIFIED CHANGES
C
6299 PRINT 1,INDEX
6300 NO=1.
6350 READ(60,26) (NINFO(M),M=1,3)
IF(EOF,60)6900,6400
6400 READ(LIN,25) (INFO(M),M=1,104)
IF(EOF,LIN) 6500,6450
C
C COMPARE A CARD AND RECORD
C
6450 IF(NINFO(1)-NO) 6500,6550,6600
C
C CARD OUT OF ORDER..STOP CHANGING ENTRIES AND COPY FILE AS IS
C
6500 REWIND LIN

```

```

REWIND LOUT
DO 6510 K=1,5
CALL SKIPFILE (LIN)
6510 CALL SKIPFILE(LOUT)
PRINT 23,INDEX
GO TO 6200

C
C SAME COUNTRY..CHANGE ENTRY
C
6550 PRINT 2,NINFO(1),NINFO(2),NINFO(3),INFO(NINFO(2))
      INFO(NINFO(2))=NINFO(3)
      READ(60,26) (NINFO(M),M=1,3)
      IF(EOF,60) 6900,6450

C
C CARD REFERS TO LARGER COUNTRY NO..WRITE RECORD
C
6600 WRITE(LOUT,25) (INFO(M),M=1,104)
      NO=NO+1.
      GO TO 6400

C
C EOF ON CHANGE DECK, WRITE REST OF INPUT FILE
C
6900 WRITE(LOUT, 25) (INFO(M),M=1,104)
      READ(LIN,25) (INFO(M),M=1,104)
      IF(EOF,LIN) 7000,6900
7000 ENDFILE LOUT
      RETURN
      END

```

```

SUBROUTINE ERROR (INDEX)
COMMON NFILE (6),INFC (500),NINFO (500),VDATA (500,2),NVDATA (50,3)
DATA (PLUS=1H+), (MINUS=1H-), (BLANK=1H )
TYPE INTEGER VDATA
LIN=1  LOUT=2  $NMAJ=17  $NVAR=70  $NDISC=3
2  FORMAT (10A8)
23  FORMAT (* FILE*,I5,*HAD A CARD OUT OF ORDER AND THE FILE WAS LEFT U
      INCHANGED..*,/,* ....DISREGARD CORRECTIONS ABOVE*)
C-----
C      THIS SUBROUTINE IS CALLED WHENEVER THERE IS A CARD OUT OF ORDER
C      ....IT ERASES ALL THE CHANGES TO THE FILE AND COPIES IT AS IT IS
C-----
      PRINT 23,INDEX
C
C      REWIND INPUT AND OUTPUT, SKIP TO FILE WHERE CARD WAS OUT OF ORDER
C      COPY ENTIRE FILE AS IS
C
      REWIND LIN
      REWIND LOUT
      NSKIP=INDEX-1
      DO 10 I=1,NSKIP
      CALL SKIPFILE (LIN)
10  CALL SKIPFILE (LOUT)
15  READ (LIN,2) (INFO(M),M=1,10)
      IF (EOF,LIN) 50,25
25  WRITE (LOUT,2) (INFO(M),M=1,10)
      GO TO 15
50  ENDFILE LOUT
C
C      SKIP OVER ANY REMAINING CHANGES FOR THIS FILE
C
75  READ 2, (NINFO(M),M=1,10)
      IF (EOF,60) 100,75
100 RETURN
      END

```

PROGRAM CODEBK

Purpose

The purpose of this program is to produce a codebook.

Method

The program reads the first five files of the dictionary tape and lists them in a readable form. As the theme and issue-variable dictionaries are read, the array JCODE is used to distinguish between non-existent theme numbers, existing theme numbers not assigned to an issue-variable, and existing theme numbers that are assigned. A list of existing themes not in an issue-variable and non-existent theme numbers is then produced from the entries in JCODE.

Dimensioned Variables

Program CODEBK contains five dimensioned variables:

- (1) NACON, dimensioned 4, contains the country number, the country name (in 2 fields), and its international abbreviation.
- (2) TITLE, dimensioned 6, contains the title of the theme, major topic, minor topic or issue-variable currently being processed.
- (3) JCODE is dimensioned 2000, each field representing a theme number (currently 1599 would be required). JCODE(K) can take on three values:
 - 0 if K is a non-existent theme
 - 1 if K exists and is not in an issue-variable
 - 2 if K exists and is in an issue-variable.
- (4) NAME, a permanent matrix of theme names dimensioned 2000 by 6, stores a 6 field title for each theme. Unlike JCODE, NAME(K, ...) contains the Kth theme title that exists.
- (5) THEMES, a temporary matrix of themes within a minor topic or issue-variable, is dimensioned 200 by 2, since no minor topic or issue-variable currently contains more than 200 themes. When the themes in minor topics are being stored, it contains the list of theme numbers in the order in which they appear in the dictionary. When the themes in issue-variables are being stored, it contains the rank and number of each theme in the order in which they are read off the dictionary tape.

Operational Effects of System Changes

Since this program's sole input is the dictionary tape, its operation is dependent upon the current format and structure of that tape. After every significant update of the dictionary, a review of the following variables is advised:

NMAJ must equal the maximum major topic number.
NVAR must equal the maximum issue-variable number.
NTHEME must equal the maximum theme number.
JCODE must be dimensioned at least as large as NTHEME.
NAME should be dimensioned as large as NTHEME by 6.
THEMES must be dimensioned at least twice as large as the maximum
number of themes in a minor topic or issue-variable.

Program Listing

The listing of program CODEBK follows.

```

PROGRAM CODERK
DIMENSION NACON(4),TITLE(6),JCODE(2000),NAME(2000,6),THEMES(200,2)
TYPE INTEGER TITLE,THEMES

```

```

C
C C INITIALIZE THE NUMBER OF MAJOR TOPICS, NUMBER OF THEMES AND THE NUMBER
C C OF VARIABLES, LOGICAL UNIT OF INPUT AND OUTPUT,...
C

```

```

LIN=1 $LOUT=61 $NMAJ=17 $NTHEME=1599 $NVAR=70
DO 50 I=1,NTHEME
50 JCODE(I)=0

```

```

C
C C READ AND WRITE COUNTRY DICTIONARY
C

```

```

WRITE(LOUT,1)
100 READ(LIN,2)(NACON(M),M=1,4)
IF(EOF,LIN)200,150
150 WRITE(LOUT,3)(NACON(M),M=1,4)
GO TO 100

```

```

C
C C READ AND WRITE THE THEME DICTIONARY
C

```

```

200 WRITE(LOUT,4)
NCOUNT=0
DO 350 INDEX=1,NTHEME
READ(LIN,5) NO,(TITLE(M),M=1,6)
JCODE(NO)=1
IF(NO)375,375,250
250 DO 300 J=1,6
300 NAME(NO,J)=TITLE(J)
WRITE(LOUT,6)NO,(NAME(NO,J),J=1,6)
350 NCOUNT=NCOUNT+1
IF(NCOUNT,GT,2000) GO TO 1000
375 CALL SKIPFILE (LIN)
WRITE(LOUT,7)NCOUNT

```

```

C
C C READ AND WRITE MAJOR AND MINOR DICTIONARY
C

```

```

DO 500 I=1,NMAJ
READ(LIN, 8)MAJOR,(TITLE(J),J=1,6),NMIN
WRITE(LOUT,9) MAJOR,(TITLE(J),J=1,6)
DO 450 II=1,NMIN
READ(LIN,10)(TITLE(J),J=1,6),NO
WRITE(LOUT,11)II,(TITLE(J),J=1,6)
IF(NO,GT,200) GO TO 900
READ(LIN,12)(THEMES(K,1),K=1,40)
DO 400 K=1,NO
KK=THEMES(K,1)
400 WRITE(LOUT,13)(NAME(KK,M),M=1,6),KK
450 CONTINUE
500 CONTINUE

```

```

      CALL SKIPFILE (LIN)
C
C      READ AND WRITE THE VARIABLE DICTIONARY
C
      WRITE(LOUT,14)
      DO 600 I=1,NVAR
      READ(LIN,15) (TITLE(J),J=1,6),NO
      IF(NO.GT.200) GO TO 950
      WRITE(LOUT,16) I, (TITLE(J),J=1,6)
      READ(LIN,17) ((THEMES(K,I),M=1,2),K=1,NO)
      DO 550 K=1,NO
      KK=THEMES(K,2)
      WRITE(LOUT,18) THEMES(K,1), (NAME(KK,J),J=1,6),KK
550   JCODE(KK)=2
600   CONTINUE
      CALL SKIPFILE (LIN)
C
C      LIST ALL THEMES NOT IN VARIABLES
C
      WRITE(LOUT,19)
      DO 700 I=1,NTHEME
      IF(JCODE(I).EQ.1) WRITE(LOUT,20) I, (NAME(I,J),J=1,6)
700   CONTINUE
C
C      READ AND WRITE LETTER DICTIONARY
C
      WRITE(LOUT,21)
800   READ(LIN,22) NO, LETTER, (TITLE(J),J=1,6)
      IF(EOF,LIN) 1050,850
850   WRITE(LOUT,23) NO,LETTER, (TITLE(J),J=1,6)
      GO TO 800
C
C      ERROR MESSAGES...OVERFLOWING DIMENSIONS
C
900   WRITE(LOUT,24) I,I,NO
      GO TO 2000
950   WRITE(LOUT,25) NO,I
      GO TO 2000
1000  WRITE(LOUT,26)
      GO TO 2000
C
C      LIST NON-EXISTENT THEME NUMBERS
C
1050  WRITE(LOUT,27)
      DO 1075 I=1,NTHEME
      IF(JCODE(I).EQ.0) WRITE(LOUT,28) I
1075  CONTINUE
C
C      FORMAT STATEMENTS
C

```



```

1 FORMAT(1H1,* COUNTRY NO          NAME          ABBREVIATION*,/)
2 FORMAT(15,2X,A8,A6,A3)
3 FORMAT(5X,13,7X,A8,A6,8X,A3)
4 FORMAT(1H1,3CX,*THEME CODES*,//,25X,* (IN NUMERICAL ORDER)*,///,4X,*C
   10DE*,17X,*THEME TITLE*,//)
5 FORMAT(15,6A8)
6 FORMAT(3X,15,*.....*,6A8)
7 FORMAT(//,* THE TOTAL NUMBER OF THEMES IS*,15,/)
8 FORMAT(15,6A8,2X,15)
9 FORMAT(1H1,5X,15,3X,6A8,//)
10 FORMAT(5X,6A8,2X,15)
11 FORMAT(//,19X,13,3X,6A8,//)
12 FORMAT(20,14)
13 FORMAT(20X,6A8,*.....*,15)
14 FORMAT(1H1,* THEME ASSIGNMENT TO VARIABLES*)
15 FORMAT(5X,6A8,2X,15)
16 FORMAT(////,30X,*VARIABLE *,13,///,20X,6A8,//)
17 FORMAT(16,15)
18 FORMAT(5X,15,3X,6A8,*.....*,15)
19 FORMAT(1H1,5X,*THE FOLLOWING THEMES ARE NOT ASSIGNED TO A VARIABLE
   1*,//)
20 FORMAT(5X,14,2X,6A8)
21 FORMAT(1H1,* LIST OF LETTERED THEMES IN NUMERICAL ORDER*,/)
22 FORMAT(15,A1,6A8)
23 FORMAT(5X,15,A1,2H*,6A8)
24 FORMAT(1X,*MAJOR TOPIC*,13,*MINOR TOPIC*,13,*HAS*,14,*THEMES WHIC
   1H IS IN EXCESS OF 200*)
25 FORMAT(1X,*THERE ARE*,15,*THEMES IN VARIABLE*,12,*WHICH EXCEEDS T
   HE 200-MAXIMUM DIMENSION*)
26 FORMAT(1X,*TOO MANY THEMES TO PROCESS...EXCEEDS 2000*)
27 FORMAT(1H1,* THE FOLLOWING THEME NUMBERS ARE UNUSED*,/)
28 FORMAT(5X,15)

```

```

C
C   END THE JOB   CENTER
C
2000 END

```

ORGANIZING THE DATA

PROGRAM MATRIX

Purpose

The primary function of this program is to convert the active data tapes into matrix form. The equivalent of a three dimensional matrix - country by theme by time period - is produced. The secondary function is to produce a retrieval printout which identifies the source of each theme.

Method

Before the matrix is created, the control cards specifying the criteria for selecting observations must be read. After the main problem card is read, the document range cards are read and NDOC is filled (see NDOC under Dimensioned Variables). Next the major topic selection card is read, if it exists, and array MAJOR is filled with a 1 when the topic is selected. Then the theme dictionary is read, and the first six fields of NAME are filled with the theme title. A row of NAME is left blank when the theme does not exist. If major topics are being selected for the retrieval printout, the major/minor dictionary is read next, and for each major topic selected, themes are given a flag of 1 in the 7th field of NAME. Lastly, the regional and lettered theme criteria for each time period is read.

The program then processes all data on a single country, filling NFREQ(theme no., time period) as observations are read. Each observation is checked against the regional, letter and document specifications and placed in NFREQ only when it meets all of the requirements. Simultaneously the retrieval printout is produced according to its specifications (i.e., all data from major topic selected, all data or only data on the matrix tape). After all the country's data from one input tape is read, all of its data from the second tape (if it exists) is read in the same manner. After all data for a country is read, NFREQ is buffered out on the tape and NFREQ is zeroed out and the next country's data is processed.

Structure of Input and Output Tapes

The primary data and secondary data tapes are each organized by country number and within country numerically by theme number. Each observation is a BCD record and the tapes are unblocked.

The matrix output tape is written in binary and organized by country. The number of records for each country depends upon the number of time periods created. The first record for a country contains the number of mentions the country has made of each of the 1599 themes during the first time period. This is followed by an equivalent record for the second time period, etc. This form is followed for all other countries.

Dimensioned Variables

There are eight dimensioned variables in program MATRIX:

(1) NDOC, dimensioned 999, has one field for each document number. NDOC(document number) equals the time period that the document number is assigned to by the document range cards or zero if it is not assigned to a time period.

(2) NAME, dimensioned 1600 by 7, is a matrix of theme titles and theme selection flags. The first six fields contain the title; the 7th contains a flag of 0 or 1, where a 1 means that the theme is assigned to a major topic selected for the retrieval output and a 0 means it is not. If major topics are not selected, all are assigned 1's. NAME(N,X) is blank whenever theme N does not exist.

(3) MAJOR, dimensioned 17, is an array of major topic flags. When major topics are selected, MAJOR(N) is 1 when topic N is selected, and is 0 otherwise. When major topics are not selected, all values are 1.

(4) NFREQ is the matrix of theme frequencies for the country being processed. It is dimensioned 1600 by 4, with one row for each theme and one column for each possible time period. This is the matrix that is written on the output tape. If major topics are selected, NFREQ is used earlier as temporary storage for themes within a minor topic.

(5) NACON, dimensioned 2, contains the two field name of the country being processed.

(6) NCLASS, dimensioned 2, is the two field classification title that is to appear at the top and bottom of each retrieval output page.

(7) MT, dimensioned 6, serves two functions. It is the temporary storage of the six field theme title before it is placed in NAME. During data processing, MT is used to store the regional flag codes for the four possible time periods. If MT(N) is 0, only non-regional remarks are included in time period N; if 1, only regional remarks; and if 2, both regional and non-regional remarks.

(8) NT, dimensioned 6, contains the letter/non-letter codes for each time period. If NT(N) is 1, only non-lettered themes are included in time period N; if 2, both letter and non-lettered themes are included; and if 3, only lettered themes are included.

Operational Effects of System Changes

This program depends upon the data tapes and the first three files of the dictionary. Assuming that their structure and format remain the same, the following changes to the dictionary files can be handled by changing these variables:

If the number of countries exceeds 149, NCOUN must be increased.

If the maximum theme number exceeds 1599, NMAX must be set equal to the maximum number and NAME and NFREQ must be redimensioned accordingly.

If the number of major topics exceeds 17, MAJOR must be redimensioned and NTOPIC must be increased.

Apart from these changes to the dictionary file, the data tape may, in the future, contain observations from document numbers larger than 999. If this occurs, MDOC and NDOC must be increased. Furthermore, if it becomes desirable to have more than 4 time periods on a single matrix tape, MT, NFREQ and NT must be redimensioned and format 1 must be changed if there are more than 8 time periods. The programmer is cautioned that expanding the number of time periods increases substantially the amount of core required.

Program Listing

The listing of program MATRIX follows.

```

PROGRAM MATRIX
DIMENSION NDOC(999),MT(6),NAME(1600,7),NCLASS(2),NACON(2)
DIMENSION MAJOR(17),NFREQ(1600,4),NT(4)
DATA (NBLANK = 1H )

C
C
C      INITIALIZE VARIABLES AND ZERO OUT NDOC

      NCOUN=149 $MDOC=999 $NTOPIC=17 $NMAX=1599
      DO 9 I=1,MDOC
9      NDOC (I) = 0

C
C
C      READ MAIN PROBLEM CARD AND DOCUMENT RANGE CARDS ...
C      FILLING IN NDOC(DOC.NO) WITH THE TIME PERIOD IT IS ASSIGNED TO
C      IT IS 0 FOR DOCUMENTS NOT SELECTED
C
      READ 1,NCARDS,NTAPES ,NRET,LDICT,LMAT,LRET,NMAJ,NTIME,(NCLASS(I),I
1=1,2)
      DO 10 J=1,NTIME
      DO 10 I=1,NCARDS
      READ 2,MIN,MAX
      DO 11 II=MIN,MAX
11      NDOC (II) = J
10      PRINT 3,MIN,MAX,J

C
C
C      READ MAJOR TOPICS TO BE INCLUDED IN THE RETRIEVAL PRINTOUT
C      IF NOT SELECTING TOPICS, ALL TOPICS WILL BE INCLUDED
C
      IF(NMAJ.NE.1) GO TO 57
      READ 7,(MAJOR(I),I=1,NTOPIC)
      GO TO 59.
57      DO 58 I=1,NTOPIC
58      MAJOR(I)=1

C
C
C      READ THEME DICTIONARY AND FILL NAME WITH 6 FIELD NAME OF THEMES

59      CALL SKIPFILE (LDICT)
      DO 80 I=1,NMAX
      READ(LDICT,81),N,(MT(L),L=1,6)
      IF(N.EQ.0) GO TO 82
      NAME(N,7)=0
      IF(NMAJ.NE.1) NAME(N,7)=1
      DO 79 J=1,6
79      NAME(N,J)=MT(J)
80      CONTINUE
82      NPAGE=1
      CALL SKIPFILE (LDICT)

C
C
C      READ MAJOR/MINOR TOPIC DICTIONARY IF SELECTING TOPICS
C      SET A FLAG IN NAME(N,7) WHEN THEME N IS ASSIGNED TO
C      A TOPIC BEING SELECTED

```

```

C      NLINE=0
      IF(NMAJ.NE.1) GO TO 601
      DO 600 I=1,NTOPIC
      READ (LDICT,22) IT
      DO 620 K=1,IT
      READ (LDICT,22) KT
      READ(LDICT,23) (NFREQ(J,1),J=1,KT)
      IF(MAJOR(I).EQ.0) GO TO 620
      DO 610 J=1,KT
      KZ=NFREQ(J,1)
610  NAME(KZ,7)=1
620  CONTINUE
600  CONTINUE
601  REWIND LDICT

C
C      READ REGIONAL SELECTION CARD FOR TIME PERIODS
C
      READ 1, (MT(I), I=1, NTIME)
      PRINT 1, (MT(I), I=1, NTIME)

C
C      READ LETTERED THEME CRITERION FOR TIME PERIODS
C
      READ 1, (NT(I), I=1, NTIME)
      PRINT 1, (NT(I), I=1, NTIME)

C
C      BEGIN PROCESSING COUNTRIES AND WRITING RETRIEVAL PRINTOUT
C
      DO 1000 I=1,NCOUN
      IF(NRET.EQ.0) GO TO 83
      READ (LDICT,61) (NACON(K),K=1,2)
      WRITE(LRET,78) (NCLASS(K),K=1,2),NPAGE, (NACON(J), J=1,2)
      NLINE=11
83  LU=50-NTAPES
      DO 50 KZ=1,NTIME
      DO 50 J=1,NMAX
50  NFREQ(J,KZ)=0

C
C      READ AN OBSERVATION
C
500 READ(LU,4) J,LY,K,L,M,N,KK,KP
      IF(EOF,LU) 700,501

C
C      ARE WE PROCESSING THIS COUNTRY DATA
C
501 IF (J-I)502,503,700
502 PRINT 5,J,N,KK,KP
      GO TO 500

C
C      YES, THEN CUMULATE THEMES IN CORRECT TIME PERIODS GIVEN INCLUSION

```

C CRITERIA

C

503 KZZ=0
DO 499 KZ=1,NTIME
IF (NDOC(N) .EQ. KZ) 489,499
489 IF (LY.EQ.MT(KZ).OR.MT(KZ).EQ.2) 490,499
490 IF (KP .EQ. 0) GO TO 492
491 IF (NT(KZ) .GE. 2) 493,499
492 IF (NT(KZ) .LE. 2) 493,499
493 NFREQ(KK,KZ)=NFREQ(KK,KZ)+1
KZZ=1
499 CONTINUE

C

C

C

WRITE OBSERVATIONS ON RETRIEVAL PRINTOUT AND SPACE PAGE

IF (NRET-1) 500,504,505
504 IF (KZZ.EQ.0) GO TO 500
505 IF (NMAJ.NE.1) GO TO 506
IF (NAME(KK,7).EQ.0) GO TO 500
506 IF (KP .EQ. 0) KP = NBLANK
IF (LY.EQ.1) 508,509
508 WRITE (LRET,511) L,M,K,N,KK,KP, (NAME(KK,KZ), KZ=1,6)
GO TO 510
509 WRITE (LRET,6) L,M,K,N,KK,KP, (NAME(KK,KZ), KZ=1,6)
510 NLINE=NLINE+1
IF (NLINE.LT.55) GO TO 500
WRITE (LRET,506) (NCLASS(IK), IK=1,2)
NPAGE=NPAGE+1
WRITE (LRET,507) (NCLASS(K), K=1,2), NPAGE, (NACON(J), J=1,2)
NLINE=5

C

C

C

RETURN TO PROCESS ANOTHER OBSERVATION FOR THIS COUNTRY

C

GO TO 500

C

C

C

RETURN TO PROCESS OBSERVATIONS FOR THE SAME COUNTRY ON THE NEXT TAPE

C

700 BACKSPACE LU
LU=LU+1
IF (LU.LE.49) GO TO 500

C

C

C

WRITE NFREQ ON MATRIX TAPE FOR COUNTRY AFTER READING ALL OF ITS DATA

DO 705 KZ=1,NTIME
BUFFER OUT (LMAT,1) (NFREQ(1,KZ), NFREQ(NMAX,KZ))
701 IF (UNIT,LMAT) 701,705
705 CONTINUE

C

C

C

COSMETIC END OF PAGE FOR RETRIEVAL OF THIS COUNTRY

```

      IF (NRET.EQ.0) GO TO 1000
      IF (NLINC.LT.55) 702,704
702 DO 703 IJ=NLINC,55
703 WRITE(LRET,701)
704 WRITE(LRET,506) (NCLASS(IK),IK=1,2)
      NPAGE=NPAGE+1

C
C      PROCESS ANOTHER COUNTRY
C
1000 CONTINUE
C
C      FORMAT STATEMENTS
C
      1 FORMAT (8I2,2A8)
      2 FORMAT (2I5)
      3 FORMAT (1X,* THE FOLLOWING DOCUMENTS ARE INCLUDED*,I5,* TO*,I5,* FO
        1R TIME PERIOD*,I5)
      4 FORMAT (I3,1X,4I1,2X,I3,6X,I4,A1)
      5 FORMAT (* CARD OUT OF ORDER*,3I5,A1)
      6 FORMAT (5X,I1,*/,I1,*/,I1,9X,I3,3X,I5,A1,3X,6A8)
      7 FORMAT (80I1)
      22 FORMAT (55X,I5)
      23 FORMAT (20I4)
      61 FORMAT (5X,A6,A6)
      78 FORMAT (1H1,4X,2A8,38X,*PAGE NO.*,I3,/,10X,*COUNTRY PROFILE FOR *,
        12A8,/,1X,*CLASSIFICATION*/,/,2X,*RELIABILITY/      THEME*,/
        2,3X,*REPETITION SOURCE NO.      THEME TITLE*,//)
      81 FORMAT (I5,6A8,I5)
      506 FORMAT (60X,2A8)
      507 FORMAT (*1*,4X,2A8,38X,*PAGE NO.*,I3,/,15X,2A8,*(CON'T)*,/)
      511 FORMAT (5X,I1,*/,I1,*/,I1,9X,I3,3X,I5,A1,3H * ,6A8)
      701 FORMAT (1X)
      ENDFILE LMAT
      IF (LRET.NE.61) ENDFILE LRET
      END

```


PROGRAM PTHEME

Purpose

The sole function of this program is to list all of the countries that have addressed specific themes and to specify their number of mentions.

Method

The program allows the user to select a set of themes for this profile by choosing ranges of theme numbers, choosing themes assigned to major topics and/or choosing themes assigned to issue-variables.

Depending on the option(s) chosen, the program reads the relevant cards and dictionary files. The array JCODE is filled with theme counters to the extent that $JCODE(\text{theme number}) = N$ when this theme number is the Nth one to be processed in this run. A maximum of 600 themes can be processed at once. An error message occurs when more than 600 are requested and the job processes the first 600 it detects. After all the themes have been assigned values in JCODE, JCODE is reordered so that the counter is numerically increasing as the theme numbers increase.

The program then begins to process the data from the matrix tape. The program buffers in one country's data and one time period at a time.

If that time period was selected, the program processes one theme at a time and only if it is a selected theme does the program fill MATRIX.

MATRIX is dimensioned 160 by 100. Each row represents a country; each column represents 6 themes, hence the 600 maximum. When an observation occurs on a theme selected, the relevant column is determined by the index of this theme number, and the value entered depends upon the location of this theme number within the column. That is, if theme number 100 is the 7th theme being processed ($JCODE(100) = 7$) and the packing factor is 6, then theme 100 is located in the first portion of the second column. Thus, the index would be 2 and the value would be $MOVE(1)$.

After all data is read off the matrix tape, MATRIX is decoded one column at a time (6 themes at a time). The 6 entries in a cell are stored in NEW. Once NEW has been filled with every country's data on the 6 themes, the titles of themes and their observations are listed. Then the program returns to process another column of data.

Dimensioned Variables

There are 11 dimensioned variables in program PTHEME:

(1) MATRIX contains the frequency of mention by each country of all themes being processed. The frequencies are packed within a column in conjunction with MOVE.

(2) MOVE, an array dimensioned 6, must be consistent with the packing factor NPACK. MOVE defines the portion of a column where frequencies of each of the 6 themes are cumulated.

(3) NAME, an array dimensioned 6, is temporary storage of the theme title being processed.

(4) JCODE, dimensioned 1600, contains indices for the themes being processed. JCODE(theme number) is zero when the theme has not been selected and N when this theme is the Nth theme being processed.

(5) NFREQ, dimensioned 1600, is temporary storage for data read off the dictionaries and the matrix tape and is also used during the decoding process. NFREQ stores the themes within the minor topics or variables when these options are used for theme selection. Later it stores all observations of a country in a particular time period. During the decoding process it stores the theme numbers being processed in such a way that $NFREQ(K) = N$, where the Kth theme being processed is theme number N.

(6) NTIME, dimensioned 4, contains a flag for the 4 time periods that could exist on the matrix tape. If $NTIME(t) = 1$, time period t's data is included in the listing; otherwise it equals 0.

(7) NCLASS, dimensioned 2, consists of the classification title to be written at the top and bottom of each page of the output.

(8) NACON, a matrix 160 by 2, contains a two field name for each of the countries in the analysis.

(9) NMAJOR, dimensioned 17, contains a flag for each major topic number. A 1 means that the themes in the major topic are to be included in the printout; a 0, that they are not.

(10) NVAR, dimensioned 70, contains a flag for each issue-variable. A 1 means that the themes in the issue-variable are to be included; a 0, that they are not.

(11) NEW is a matrix dimensioned 160 by 6. Each row represents a country and each column a theme number. NEW is the decoded form of a single column of MATRIX in which the entries have been unpacked. An entry is the frequency with which a country has made this remark under the conditions specified on the main problem card.

Operational Effects of System Changes

This program is dependent upon the matrix tape and the first four files of the dictionary tape. If minor modifications to the four files of the dictionary are made, the programmer is cautioned to change the following variables in this program:

If the maximum theme number should exceed 1599, NMAX must be set equal to the maximum number and JCODE and NFREQ must be dimensioned at least as large.

If the number of countries exceeds 149, NCOUN must be increased. If NCOUN is increased beyond 160, MATRIX, NACON and NEW must be increased accordingly.

If the number of major topics exceeds 17, NTOPIC and NMAJOR must be increased.

If the number of issue-variables exceeds 70, NVARB and NVARS must be increased.

If modifications are made to the matrix tape, such as increasing the number of time periods that can exist, NTIME must be dimensioned at least as large as the maximum number of time periods, and the format and read statements for the main problem card must be changed accordingly.

Apart from these tape changes, the programmer is cautioned to make certain that the number of observations by a single country on a theme under the criteria established on the main problem card never exceeds 128. If it does, the packing factor must be reduced, MOVE and NEW would have to be redimensioned, and the values of MOVE would have to be changed accordingly.

Program Listing

The listing of program PTHEME follows.

```

PROGRAM PTHEME
DIMENSION MATRIX(160,100),NAME(6),MOVE(6),JCODE(1600),NFREQ(1600)
DIMENSION NTIME(4),NCLASS(2),NACON(160,2),NEW(160,6)
DIMENSION NMAJOR(17),NVAR(70)

C
C INITIALIZE MAX NO OF THEMES, COUNTRIES, MAJOR TOPICS, VARIABLES AND
C COLUMNS THAT FIT IN THE CURRENT DIMENSIONS...ZERO OUT COUNTER AND JCODE
C ESTABLISH THE PACKING FACTOR
C
NPACK=6 $NTOPIC=17 $NVAR=70 $NMAX=1599 $NCOUN=149
MOVE(1)=2**40 $MOVE(2)=2**32 $MOVE(3)=2**24 $MOVE(4)=2**16
MOVE(5)=2**8 $MOVE(6)=1 $ KQ=0 $NCOL=100
DO 121 I=1,NMAX
121 JCODE(I) = 0

C
C READ MAIN PROBLEM CARD. THEN READ THEME RANGE SELECTION
C CARDS FILLING JCODE(THEME NO) WITH THE COUNTER ON THE THEME
C
READ 1,LU,LOUT,LDICT,NRANGE,MTIME,(NTIME(I),I=1,4),(NCLASS(K),K=1,
12) , NMAJ, NVAR
DO 11 J=1,NRANGE
READ 1,MIN,MAX
DO 10 I=MIN,MAX
KQ = KQ + 1
IF (KQ.GE. 600) GO TO 122
10 JCODE(I)=KQ
11 CONTINUE

C
C READ COUNTRY DICTIONARY AND SKIF TO MAJOR/MINOR DICTIONARY
C
122 READ(LDICT,4)((NACON(I,J),J=1,2),I=1,NCOUN)
CALL SKIPFILE (LDICT)
CALL SKIPFILE (LDICT)

C
C IF SELECTING THEMES IN MAJOR TOPICS, READ DICTIONARY AND FILL IN
C JCODE(THEME NO) WITH THE COUNTER ON THE THEMES IN THE TOPICS SELECTED
C
IF(NMAJ .EQ. 0) GO TO 1100
READ (60,1111) (NMAJOR(I),I=1,NTOPIC)
DO 1112 I=1,NTOPIC
READ (LDICT, 1122) KMAJ
DO 1620 K=1,KMAJ
READ (LDICT,1122) KMIN
READ (LDICT,1123) (NFREQ(I), II=1,KMIN)
IF (NMAJOR(I) .EQ. 0) GO TO 1620
DO 1610 J=1,KMIN
KZ = NFREQ(J)
IF (JCODE(KZ) .NE. 0) GO TO 1610
KQ = KQ + 1
JCODE(KZ)=KQ

```

```

        IF (KQ.GE. 600) GO TO 12
1610 CONTINUE
1620 CONTINUE
1112 CONTINUE
C
C   IF SELECTING THEMES IN VARIABLES, READ DICTIONARY AND FILL IN
C   JCODE(THEME NO) WITH COUNTER OF THEMES IN VARIABLES SELECTED
C
1100 CALL SKIPFILE (LDICT)
    IF (NVAR .EQ. 0) GO TO 12
    READ (60, 1111) (NVAR (I), I=1, NVAR)
    DO 1212 I=1, NVAR
    READ (LDICT, 1122) KVAR
    READ (LDICT, 1221) (NFREQ(IK), IK =1, KVAR)
    IF (NVAR(I) .EQ. 0) GO TO 1212
    DO 1230 J = 1, KVAR
    KZ = NFREQ (J)
    IF (JCODE (KZ) .NE. 0 ) GO TO 1230
    KQ = KQ + 1
    JCODE(KZ)=KQ
    IF (KQ.GE. 600) GO TO 12
1230 CONTINUE
1212 CONTINUE
12 REWIND LDICT
    CALL SKIPFILE (LDICT)
C
C   REORDER JCODE FOR ALL THEMES SELECTED SUCH THAT THE COUNTER IS
C   NUMERICALLY INCREASING AS THE THEME NUMBERS INCREASE
C
    K = 1
    DO 2000 I=1, NMAX
    IF (JCODE (I) .EQ. 0 ) GO TO 2000
    JCODE (I) = K
    K = K + 1
2000 CONTINUE
C
C   READ ALL DATA ON A COUNTRY IN A GIVEN TIME PERIOD
C   IF NOT A TIME PERIOD SELECTED, READ THE NEXT TIME PERIOD OR COUNTRY
C   IF IT IS, AND ITS A THEME TO BE PROCESSED
C   FILL MATRIX(COUNTRY, PACKING FACTOR) WITH THE PACKING WEIGHT
C
    IF (KQ .GE. 600) PRINT 3
    NCOL = (KQ / NPACK) + 1
    DO 999 IK=1, NCOUN
    DO 998 JK=1, MTIME
15 BUFFER IN(LU, 1) (NFREQ(1), NFREQ(NMAX))
20 IF (UNIT, LU) 20, 16, 1000
16 IF (NTIME (JK).EQ.0) GO TO 998
    DO 100 I=1, NMAX
    IF (NFREQ(I).EQ.0) GO TO 100

```

```

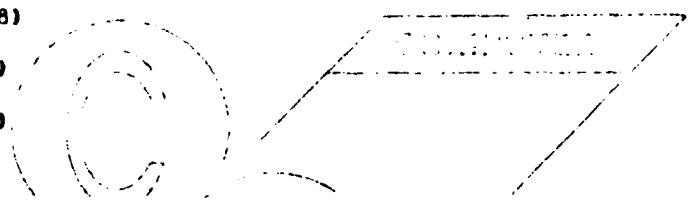
IF(JCODE(I).EQ.0) GO TO 100
II=((JCODE(I)-1)/NPACK)+1
KI=(II-1)*NPACK
DO 50 KK=1,NPACK
KI=KI+1
IF(JCODE(I).EQ.KI) 49,50
49 MATRIX(IK,II)=MATRIX(IK,II)+(MOVE(KK)*NFREQ(I))
50 CONTINUE
100 CONTINUE
998 CONTINUE
999 CONTINUE
C
C AFTER READING ALL COUNTRIES AND TIME PERIODS, DECODE MATRIX ONE SET
C OF THEMES AT A TIME...AND WRITE OUT THE OBSERVATIONS WITH THE THEME TITLE
C
1000 K=1
DO 450 I=1,NMAX
IF(JCODE(I).EQ.0) GO TO 450
NFREQ(K)=I
K=K+1
450 CONTINUE
NLINE=0
DO 500 J=1,NCOL
KTHEME=((J-1)*NPACK)+1
MTHEME=KTHEME+(NPACK-1)
DO 499 I=1,NCCUN
DO 451 LL=1,NPACK
451 NEW(I,LL)=0
IF(MATRIX(I,J).EQ.0) GO TO 499
L=0
DO 400 KK=1,NPACK
NEW(I,KK)=(MATRIX(I,J)-L)/MOVE(KK)
L=L+(NEW(I,KK)*MOVE(KK))
400 CONTINUE
499 CONTINUE
DO 460 LC=1,NPACK
KK=(KTHEME-1)+LC
461 READ(LDICT,6) NO,(NAME(K),K=1,6)
IF(NO-NFREQ(KK))461,470,480
470 WRITE(LOUT,7) NO,(NAME(K),K=1,6)
NLINE=NLINE+2
DO 465 I=1,NCCUN
IF(NEW(I,LC).EQ.0) GO TO 465
WRITE(LOUT,8) (NCON(I,N),N=1,2),NEW(I,LC)
NLINE=NLINE+1
IF(NLINE.GT.55) 419,465
418 WRITE(LOUT,9) (NCLASS(N),N=1,2)
WRITE(LOUT,18) (NCLASS(N),N=1,2)
NLINE=1
465 CONTINUE

```

GO TO 460
480 BACKSPACE LDICT
460 CONTINUE
500 CONTINUE

C
C
C

FORMAT STATEMENTS

1 FORMAT (5I5,4I1,2A8,2I5)
2 FORMAT (30I4)
3 FORMAT (1X,* ONLY THE FIRST 600 THEMES ARE INCLUDED IN THIS RUN*)
4 FORMAT (5X,A8,A6)
6 FORMAT (I5,6A8)
7 FORMAT (2X,I5,2X,6A8,/) 
8 FORMAT (10X,A8,A6,I5)
9 FORMAT (61X,2A8)
18 FORMAT (1H1,2X,2A8)
1111 FORMAT (70I1)
1122 FORMAT (55X, I5)
1123 FORMAT (20I4)
1221 FORMAT (8(5X,I5)).

C

END

PROGRAM LETTER

Purpose

This program is a special case of program PTHEME and handles data on lettered themes. The sole function of LETTER is to list all the countries that have addressed particular lettered themes and to specify the number of mentions. A theme profile is produced for all lettered themes that were selected.

Method

Unlike PTHEME, this program reads the original data tapes. First, the main problem card is read and then the document numbers to be included in this printout are read from the document range cards. The array NDOCS is filled so that NDOCS(N) is 1 when observations from document N are to be included and 0 when they are not,

The lettered theme dictionary is read and the first 315 theme numbers with letters are identified. JCODE(theme no.) is assigned a disc number where this theme's data will be temporarily stored. Currently seven theme numbers, each with its various letter designates, are assigned to a disc unit 1 to 45, hence the 315 maximum.

Then the input data tape(s) are read an observation at a time. If the observation is on a lettered theme being processed in this run and it meets the document and regional code criteria, it is written on the disc unit that stores this theme number (and 6 others).

After all data has been read from the tape(s) and the appropriate observations have been stored on disc, the lettered theme dictionary is reread one theme number and letter at a time. The appropriate disc is searched for observations and as they are found they are written on the printout. The disc is then rewound and the next lettered theme is processed in the same manner.

Dimensioned Variables

Program LETTER has five dimensioned variables:

(1) NDOCS is dimensioned 999 with one field for each existing document number. If document N's data is to be included in the printout, NDOCS(N) = 1; otherwise it equals 0.

(2) NACON, a matrix dimensioned 160 by 2, contains a two field name for each country.

(3) JCODE is an array 1599 long with one field for each theme number. Beginning with the smallest theme number for this run as specified on the main problem card, JCODE is filled with a counter for each theme number with letters; that is, JCODE(theme number) = N whenever the theme number has lettered versions to it and it is the Nth such theme number found. JCODE is zero in all other cases.

(4) NCOUNT, an array dimensioned 45, one field for each temporary disc to be used, is merely the number of observations that exists on each disc.

(5) MT, dimensioned 6, contains a six field title of the theme being processed.

Operational Effects of System Changes

The normal operation of this program depends upon the current structure and format of the data tape, country dictionary, and lettered theme dictionary. The following minor modifications can be handled by changing the following variables:

If the number of documents exceeds 999, NDOCS will have to be redimensioned.

If the maximum theme number exceeds 1599, MAXT and JCODE must be at least as large as the maximum number.

If the number of countries exceeds 160, both MAXC and NACON must be increased.

This program will currently process 315 theme numbers with all of their lettered versions (i.e., 45 discs with 7 per disc). This handles all existing lettered theme numbers. If, in the future, it does not, NPACK could be increased and then more themes would be written on a single disc.

Program Listing

The listing for program LETTER follows.

```

PROGRAM LETTER
DIMENSION NCCUNT(45),MT(6),NDOCS( 999),NACON(160,2),JCODE(1599)
DATA (NBLANK =1H )

C
C INITIALIZE MAX NO OF THEMES,COUNTRIES AND DISCS THAT CAN BE RUN
C WITH THE CURRENT DIMENSIONS..SET NSKIP TO BE THE NO OF FILES TO GET TO
C LETTERED THEME DICTIONARY AND ESTABLISH THE PACKING FACTOR..
C ZERO OUT COUNTERS,NCCUNT AND JCODE
C
MAXT=1599 $MAXC=160 $MAXD=45 $NSKIP=4 $LCOUNT=0 $NOISC=0
NPACK = 7
DO 10 I = 1,MAXT
10 JCODE(I) = 0
DO 15 I=1,MAXD
15 NCCUNT(I) = 0

C
C READ MAIN PROBLEM CARD,COUNTRY DICTIONARY AND DOCUMENT SELECTION CARDS
C SET UP NDOC WITH 1'S FOR DOCUMENTS SELECTED IN THIS RUN
C
READ 1, NTAPES, NTHEME, NDOC, NPEG, LDICT, LOUT, NCOUN
READ(LDICT,6) ((NACON(I,J),J=1,2), I=1,NCCUN)
DO 18 I=1,NDOC
READ 1, MIN, MAX
DO 17 J=MIN, MAX
17 NDOCS(J) = 1
18 CONTINUE

C
C AFTER SKIPPING TO LETTERED THEME DICTIONARY..READ ONE THEME AT A TIME
C IF IT IS TO BE PROCESSED,FILL JCODE(THEME NO) WITH PACKING FACTOR
C WHICH REFERENCES DISC WHERE THE THEME WILL BE STORED
C
DO 20 I=1,NSKIP
20 CALL SKIPFILE (LDICT)
50 READ (LDICT,2) (M,N, (MT(I),I=1,6))
IF (EOF,LDICT) 100,51
51 IF (M .LT. NTHEME) GO TO 50
IF (JCODE (M) .NE. 0) GO TO 50
JCODE (M) = (LCOUNT/NPACK) + 1
LCOUNT = LCOUNT + 1
LL = ((LCOUNT + 1) / NPACK) + 1
IF(LL .GT. MAXD) 60,50
60 PRINT 3, M,N, (MT(I),I=1,6)
100 REWIND LDICT
DO 101 I=1,NSKIP
101 CALL SKIPFILE (LDICT)

C
C READ FIRST TAPE..ONE OBSERVATION AT A TIME..
C IF OBSERVATION MEETS THE CRITERION FOR REGIONAL CODE,DOCUMENT
C CODE AND LETTERED THEME,WRITE THE OBSERVATION ON APPROPRIATE DISC
C

```

```

    LU = 50-NTAPES
150 READ (LU,4) J,K,L,M,N
    IF (EOF,LU) 1000,151
151 IF (K.EQ. NREG .OR. NREG.EQ. 2) 152,150
152 IF (NDOCS (L) .EQ. 0) GO TO 150
    IF (N.EQ.NBLANK .OR. JCODE(M) .EQ.0) GO TO 150
    KK = JCODE(M)
    WRITE (KK,5) J,M,N,L
    NCOUNT (KK) = NCOUNT (KK) + 1
    IF (KK.GT. NDISC) NDISC = KK
    GO TO 150

C
C IF MORE THAN ONE INPUT TAPE, RETURN TO READ ALL OF ITS OBSERVATIONS AS
C DESCRIBED EARLIER
C IF NOT...ENDFILE AND REWIND ALL DISC UNITS
C
1000 LU = LU+1
    IF (LU.LE. 49) GO TO 150
    DO 1100 J = 1,NDISC
    ENDFILE J
1100 REWIND J

C
C REREAD LETTERED THEME DICTIONARY, ONE LETTER AT A TIME
C WRITE OUT THEME NUMBER/LETTER AND TITLE AND SEARCH THE
C APPROPRIATE DISC FOR ALL OF ITS OBSERVATIONS AND WRITE ANY FOUND
C
2000 READ (LDICT,2) (M,N,(MT(I),I=1,6))
    IF (EOF,LDICT) 5000,2001
2001 NREC = NCOUNT(JCODE(M))
    IF (JCODE (M) .EQ.0) GO TO 2000
    WRITE (LOUT,7) (M,N, (MT(I),I=1,6))
    DO 2500 I=1,NREC
    KK = JCODE (M)
    READ (KK,5) JJ,MM,NN,LL
    IF (MM.NE. M .OR. NN.NE. N) GO TO 2500
    WRITE (LOUT,8) (NACON (JJ,K), K=1,2),LL
2500 CONTINUE

C
C REWIND DISC AND RETURN TO READ THE NEXT LETTERED THEME
C
    REWIND KK
    GO TO 2000

C
C FORMAT STATEMENTS
C
1 FORMAT (16I5)
2 FORMAT (I5, A1, 6A8)
3 FORMAT (1H1, * THE FOLLOWING THEME WAS THE LAST TO BE PROCESSED*,
1I5,A1,2X,6A8)
4 FORMAT (I3, I2, 5X, I3, 6X, I4, A1)
5 FORMAT (2I5,A1,I5)
6 FORMAT(5X,A8,A6)
7 FORMAT (/, 5X, I4, A1, 2X, 6A8)
8 FORMAT(10X,A8,A6,2X,I5)

C
5000 END

```

PROGRAM PROFILE

Purpose

This program and its subroutine produce one of four possible reports of the data on the matrix tape. Depending on the option selected, the program lists in numerical order:

- countries with their frequency of mention of themes by time period,
- countries with their data organized into major and minor topics by time period,
- themes with the number of mentions by all countries during each time period,
- countries with their data organized into selected issue-variables for each time period.

The last option, country profile by issue-variable, is the only one that continues the flow. It produces an output tape which converts the matrix input into observations on issue-variables for use by program SCORES. Up to 45 issue-variables can be processed at once.

Method

When producing any of the above four reports, the program reads the main problem card and then all selection cards in the following order. First, the country range selection cards, if any, are read and NACON(country number, 3) is set to 1 if the country is to be included and 0 if not. If there is no country selection card, all countries receive a 1. The theme range selection cards are read and JCODE(theme number) is set to 1 if the theme is selected and 0 if not. The same procedure as with country selection cards is then followed for the major topic selection cards and issue-variable selection cards, if they exist, and NTOPIC. But a country profile by issue-variable cannot have been requested when topic selection cards exist, and vice versa. Then the final control card, titles of time periods, is read, storing only those titles used in this job.

The dictionary files are read next, first the country dictionary and then the theme dictionary. If major topics are being selected, this dictionary is read, the titles are stored in NAMEM, and the themes within selected topics are stored on disc units. The same procedure is followed for reading the issue-variable dictionary when issue-variables are being selected and a country profile by issue-variable has been requested.

At this point, the type of profile requested will determine the remaining logic to follow. If the selection is:

(1) A country profile with theme frequency, all time period data on the matrix tape for a given country is buffered into NFREQ(theme number, time period). If this country has been selected as one to process, the themes are taken one at a time in numerical order.

If a theme is one that was selected and this country has data on it, this theme is listed with this country's frequency of mentions in each of the selected time periods. Once all themes have been processed, the next country's data is buffered in.

(2) A country profile by major/minor topic, all existing time period data for a given country is buffered into NFREQ(theme number, time period). If this country has been selected as one to process, the major topics are processed one at a time. For each selected major topic, minor topic themes are read off the disc. If this country has data on any of these themes, the theme is listed with the frequency of mention in each of the selected time periods. Once all minor topics from selected major topics have been processed in this manner, the next country's data is buffered in.

(3) A theme profile with total number of mentions, all data from the tape (for selected countries) is buffered into NFREQ(theme number, time period) and cumulated. The selected themes are listed in numerical order with number of mentions across all countries in the selected time periods.

(4) A country profile by issue-variable, all time period data for a given country is buffered into NFREQ(theme number, time period). If the country has been selected, the issue-variables are taken one at a time. (A maximum of 45 variables can be processed.) For each selected issue-variable, themes are read from the disc. If the country has data on these themes, they are listed with the frequency of mention in each of the selected time periods. Simultaneously, each mention of a theme within a selected issue-variable is buffered out to the output tape. (See Structure of Issue-Variable Data Output Tape below.) Once all selected issue-variables have been processed, the next country's data is buffered in.

All four types of profile call subroutine HEADER, which prints the classification on the bottom of each retrieval output page and a continuation heading on the top of the next page.

Structure of Issue-Variable Data Output Tape

This binary output tape is created whenever a country profile by issue-variable is requested. This tape is necessary to continue the flow through the remaining programs. It contains all the data listed in the country profile printout, but each mention of a theme is written on a separate record. It is organized by country and within country by issue-variable. Within an issue-variable the observations on themes are listed in the order in which they appear in the variable, with the first time period's data on a given theme appearing before the second, etc. A record has the following field interpretations:

- 1) country number
- 2) issue-variable number
- 3) theme number
- 4) rank of this theme in this issue-variable
- 5) time period of this observation.

Dimensioned Variables

There are 15 dimensioned variables in this program:

(1) NACON is dimensioned 160 by 3, with one row for each country. The first two columns contain the country name and the third serves as a flag for selecting countries. If it is 1, the country is processed; if it is 0, the country is skipped.

(2) JCODE, dimensioned 1800, contains theme number flags. If JCODE(N) is 1, theme number N exists and is included in the analyses. If JCODE(N) is 0, either it does not exist or it is excluded from the analysis.

(3) NTOPIC, dimensioned 70, contains flags for either selected topics or selected issue-variables, depending on the type of profile requested. If NTOPIC(N) is 1, topic or variable N is included; if 0, it is excluded from the analysis.

(4) MT, 6 fields long, is the temporary storage of the theme title being processed prior to its being stored in NAME.

(5) NAME, dimensioned 1800 by 6, is a matrix of theme titles. NAME(N, 1-6) contain the title for theme number N.

(6) NAMEM, dimensioned 170 by 5, is a matrix of titles of either topics or issue-variables, depending on the type of profile requested. If it contains topic titles, they appear in the order they are read, i.e., major topic 1 followed by its various minor topics, then major topic 2, etc. If it contains issue-variable titles, they appear in numerical order. In either case, the first four columns are the title and the fifth is the number of minor topics in the major topic or themes in a minor topic or variable.

(7) KTHEME, a matrix 170 by 2, is the temporary storage of theme numbers and their ranks in the issue-variable being processed. In each case, the rank fills the first column and the theme number the second,

(8) KNT, dimensioned 17, contains a counter that locates the beginning of major topics in the matrix NAMEM. If KNT(3) = 10, the major topic 3 title appears in the 10th row of NAMEM followed by its minor topic titles.

(9) NCLASS is a two field array of the classification to be written at the top and bottom of every page of the printout.

(10) NFREQ is the working matrix of frequency of mention of a theme number in a given time period. It is dimensioned 1800 by 4 with a row for each theme and a column for each of 4 possible time periods. For country profiles, each cell contains the frequency for the country being processed. For a theme profile, it contains the cumulative frequency for all countries.

(11) NTIME, dimensioned 4, contains flags for the time periods selected. Originally, it is read as a series of 1's and 0's so that NTIME(N) = 1 when the Nth time period is selected. During most of the program, NTIME(N) = J, which means that the Nth time period being processed is the Jth time period on the input tape.

(12) FLAG, dimensioned 20, contains a 0 or 1 flag for each minor topic of the major topic being processed. If FLAG(N) is 0, minor topic N's title has not been written on the printout; a 1 means that it has been.

(13) NMAJOR, dimensioned 90, contains the temporary storage of the theme numbers within the minor topic being processed.

(14) NAMET, dimensioned 4, is an array of time period titles for all the time periods on the input tape. After they have been read, the titles are consolidated so that NAMET contains only the titles of selected time periods and they appear in the first available cell; i.e., NAMET(N) contains the title of the Nth time period that has been selected.

(15) MCOUNT, dimensioned 1800, is the temporary storage of frequency of mentions by a country in a time period. It is used when a theme profile is produced. This array is later cumulated in NFREQ.

Operational Effects of System Changes

Program PROFLE depends upon the matrix tape and the first four files of the dictionary tape. Assuming that their structure and format remain the same, the following changes to the dictionary files can be handled by changing these variables:

If the number of countries exceeds 149, NCOUN must be increased. If it exceeds 160, NACON must be redimensioned.

If the maximum theme number exceeds 1599, NMAX must be set equal to the new maximum number. If this new maximum should exceed 1800, JCODE, NAME, NFREQ, and MCOUNT must be redimensioned.

If the number of major topics exceeds 17, NMAJ must be changed and KNT must be redimensioned. If it exceeds 70, NTOPIC must be increased.

If the total number of major and minor topics ever exceeds 170, NAMEM must be increased.

If the number of minor topics in a major topic ever exceeds 90, NMAJOR must be increased.

If there are more than 20 minor topics in a major topic, FLAG must be redimensioned.

If the number of issue-variables exceeds 70, NVAR must be increased and NTOPIC redimensioned. If it should exceed 170, NAMEM must be redimensioned.

If the number of themes in an issue-variable ever exceeds 170, KTHEME must be redimensioned.

Apart from these changes to the dictionary files, the matrix tape could in the future contain more than 4 time periods. If this occurs, NFREQ, NTIME, and NAMET must be increased.

Finally, if it becomes desirable to process more than 45 issue-variables at once, the logic of this program will have to be changed. Currently, each issue-variable is assigned to a separate disc and only 45 discs are available.

Program Listing

The program and subroutine listings follow.

PROGRAM PROFILE

COMMON NACON(160,3),NCLASS(2)
 DIMENSION JCODE(1800),NTOPI(70),MT(6),NAME(1800,6),NTIME(4)
 DIMENSION NAMEM(170,5),KTHEME(170,2),KNT(17),NFREQ(1800,4)
 DIMENSION FLAG(20),NMAJOR(90),MCOUNT(1800),NAMET(4)
 TYPE INTEGER FLAG

C
 C INITIALIZE THE NUMBER OF COUNTRIES, THEMES, TOPICS AND VARIABLES
 C SET 61 TO BE THE OUTPUT FOR THE PROFILE PRODUCED ..
 C READ MAIN PROBLEM CARD

NCOUN=149 \$NMAX=1599 \$LOUT=61 \$NMAJ=17 \$NVAR=70 \$NPAGE=0
 READ 1,LU,LDICT,LOUTT,ICOUN,ITOPIC,LCOUN,LTHEME,LTOPI(NTIME
 1E(J),J=1,4),(NCLASS(I),I=1,2)
 WRITE(LOUT,25)LU,LDICT,LOUT,ICOUN,ITOPIC

C
 C IF SELECTING COUNTRIES, READ COUNTRY RANGE SELECTION CARDS AND
 C FILL NACON(COUNTRY NO,3) WITH A 1 WHEN SELECTED..0 IF NOT
 C IF NOT SELECTING COUNTRIES ALL CELLS ARE 1.

IF(LCOUN.EQ.0) GO TO 35
 DO 30 I=1,LCOUN
 READ 3,MIN,MAX
 WRITE(LOUT,2) MIN,MAX
 DO 29 J=MIN,MAX
 29 NACON(J,3)=1
 30 CONTINUE
 GO TO 40
 35 WRITE(LOUT,4)
 DO 31 I=1,NCOUN
 31 NACON(I,3)=1

C
 C IF SELECTING THEMES, READ THEME SELECTION CARDS AND FILL
 C JCODE(THEME NO) WITH 1 IF SELECTED AND 0 IF NOT.
 C IF NOT SELECTING THEMES, JCODE IS ALWAYS 1

40 IF(LTHEME.EQ.0) GO TO 55
 DO 50 I=1,LTHEME
 READ 3,MIN,MAX
 WRITE(LOUT,5) MIN,MAX
 DO 49 J=MIN,MAX
 49 JCODE(J)=1
 50 CONTINUE
 GO TO 60
 55 WRITE(LOUT,6)
 DO 56 I=1,NMAX
 56 JCODE(I)=1

C
 C IF SELECTING MAJOR TOPICS OR VARIABLES, READ MAJOR TOPIC OR
 C VARIABLE SELECTION CARD AND FILL NTOPI(NO) WITH 1 IF SELECTED

```

C      AND 0 IF NOT. IF NOT SELECTING TOPICS OR VARIABLES OR AN INCOMPAT-
C      ABLE REQUEST IS MADE, NTOPIC IS ALL 1.
C
60 IF (LTOPIC-1) 90,61,70
61 IF (ICOUN.NE.2) GO TO 89
   READ 8, (NTOPIC(I), I=1, NMAJ)
   WRITE (LOUT, 8) (NTOPIC(I), I=1, NMAJ)
   GO TO 92
70 IF (ICOUN.NE.3) GO TO 89
   READ 8, (NTOPIC(I), I=1, NVAR)
   WRITE (LOUT, 8) (NTOPIC(I), I=1, NVAR)
   GO TO 92
89 WRITE (LOUT, 7) ICOUN, LTOPIC
   STOP
90 WRITE (LOUT, 18)
   DO 91 I=1, NVAR
91 NTOPIC(I)=1

C
C      THE TITLES OF THE TIME PERIODS ARE READ AND REARRANGED
C
92 READ 34, ((NAMET(I), I=1, MTIME))
   KTIME=0
   DO 93 I=1, MTIME
93 KTIME=NTIME(I)+KTIME
   DO 94 I=1, KTIME
     J=I
     IF (NTIME(I).EQ.0) 95,94
95 J=J+1
     IF (NTIME(J).EQ.0) 95,96
96 NTIME(J)=0
     NAMET(I)=NAMET(J)
94 NTIME(I)=J

C
C      READ COUNTRY DICTIONARY
C
   READ (LDICT, 19) ((NACON(I, J), J=1, 2), I=1, NCOUN)
   CALL SKIPFILE (LDICT)

C
C      READ THEME DICTIONARY FILLING JCODE (THEME NO) WITH 1 IF SELECTED
C      AND IT EXISTS...0 OTHERWISE
C
   DO 120 I=1, NMAX
   READ (LDICT, 21) N, (MT(K), K=1, 6)
   IF (N) 121, 121, 105
105 DO 110 J=1, 6
110 NAME(N, J)=MT(J)
120 IF (JCODE(N).NE.0) JCODE(N)=-1
121 CALL SKIPFILE (LDICT)
   DO 124 I=1, NMAX
   IF (JCODE(I)) 130, 124, 132

```

```

130 JCODE(I)=1
    GO TO 124
132 JCODE(I)=0
124 CONTINUE
C
C IF SELECTING MAJOR TOPICS AND PRODUCING A COUNTRY PROFILE BY MAJOR/
C MINOR, READ MAJOR/MINOR DICTIONARY AND STORE NAMES OF TOPICS IN
C NAMEM AS THEY ARE READ..ZERO OUT JCODE FOR ALL THEMES NOT INCLUDED
C IN THE TOPICS SELECTED...FOR TOPICS SELECTED, WRITE ALL THEMES IN
C EACH MINOR TOPIC ON THE DISC UNITS REPRESENTING EACH SELECTED TOPIC
C
    IF(LTOPIC.EQ.1) GO TO 123
    IF(ICOUN.NE.2) GO TO 300
123 L=0
    DO 220 I=1,NMAJ
        L=L+1
        READ(LDICT,22)(NAMEM(L,KP),KP=1,5)
        KZ=NAMEM(L,5)      $KNT(I)=L
        DO 210 J=1,KZ
            L=L+1
            READ(LDICT,22)(NAMEM(L,KP),KP=1,5)
            NMINOR=NAMEM(L,5)
            READ(LDICT,23)(NMAJOR(LK),LK=1,NMINOR)
            IF(NTOPIC(I).EQ.0) 202,205
202 DO 203 LK=1,NMINOR
        KK=NMAJOR(LK)
203 JCODE(KK)=0
        GO TO 210
205 WRITE(I)(NMAJOR(LK),LK=1,NMINOR)
210 CONTINUE
220 ENDFILE I
C
C IF SELECTING VARIABLES AND PRODUCING A COUNTRY PROFILE BY VARIABLE,
C READ VARIABLE DICTIONARY AND STORE TITLES IN NAMEM
C AND THE THEMES WITH THEIR RANKS ARE STORED ON DISC.
C
300 IF(LTOPIC.EQ.2) GO TO 221
    IF(ICOUN.NE.3) GO TO 489
221 NOISC=1
    CALL SKIPFILE (LDICT)
    DO 330 I=1,NVAR
        READ(LDICT,22)(NAMEM(I,J),J=1,5)
301 L=NAMEM(I,5)
        READ(LDICT,20)((KTHEME(K,J),J=1,2),K=1,L)
        IF(NTOPIC(I).NE.1) 330,305
305 WRITE(NOISC)((KTHEME(K,J),J=1,2),K=1,L)
        ENDFILE NOISC
        NOISC=NOISC+1
330 CONTINUE
C

```

```

C      SKIP TO APPROPRIATE TYPE OF PROFILE REQUESTED
C
489 IF(ITOPIC-1) 490,3999,6000
490 IF(ICCUN-2)499,2500,3000
C-----
C COUNTRY PROFILE LISTING COUNTRIES WITH THE FREQUENCY OF THEIR REMARKS
C-----
499 DO 1000 I=1,NCOUN
      DO 501 L=1,MTIME
      BUFFER IN(LU,1)(NFREQ(1,L),NFREQ(NMAX,L))
500 IF(UNIT,LU)500,501
501 CONTINUE
      IF(NACON(I,3))1000,1000,502
502 NPAGE=NPAGE+1
      WRITE(LOUT,14) (NCLASS(K),K=1,2),NPAGE,(NACON(I,J),J=1,2),(NAMET(L)
1,L=1,KTIME)
      NLINE=NLINE+1
      DO 550 K=1,NMAX
      DO 551 LL=1,MTIME
551 NFREQ(K,LL)=NFREQ(K,LL)*JCODE(K)
      L=0
      DO 552 LL=1,KTIME
      KZ=NTIME(LL)
      NFREQ(K,LL)=NFREQ(K,KZ)
      IF(NFREQ(K,LL).NE.0) L=1
552 CONTINUE
      IF(L.EQ.0) GO TO 550
503 WRITE(LOUT,15) K,(NAME(K,L),L=1,6),(NFREQ(K,LL),LL=1,KTIME)
      NLINE=NLINE+1
      IF(NLINE.GE.55) CALL HEADER(NLINE,NPAGE,I,LOUT,ITOPIC)
550 CONTINUE
      CALL HEADER(NLINE,NPAGE,I,LOUT,ITOPIC)
1000 CONTINUE
      GO TO 6000
C-----
C COUNTRY PROFILE BY MAJOR AND MINOR CATEGORIES
C-----
2500 DO 2600 I=1,NCOUN
      DO 2501 L=1,MTIME
      BUFFER IN(LU,1)(NFREQ(1,L),NFREQ(NMAX,L))
2502 IF(UNIT,LU)2502,2501
2501 CONTINUE
2561 IF(NACON(I,3))2600,2600,2562
2562 NPAGE=NPAGE+1
      WRITE(LOUT,32) (NCLASS(K),K=1,2),NPAGE,(NACON(I,KK),KK=1,2),(NAMET(
1J),J=1,KTIME)
      NLINE=7
      DO 2575 J=1,NMAJ
      JJ=KNT(J)
      IF(NTOPIC(J)) 2575,2575,2563

```

```

2563 WRITE(LOUT,12) J, (NAMEM(JJ,L),L=1,4)
      NLINE=NLINE+2 $REWIND J $NMINCR=NAMEM(JJ,5)
      DO 2570 II=1,NMINOR
      FLAG(II)=0 $L=JJ+II $KL=NAMEM(L,5)
      READ(J)(NMAJOR(LK),LK=1,KL)
2565 DO 2564 LK=1,KL
      LZ=NMAJOR(LK)
      DO 2571 LL=1,MTIME
2571 NFREQ(LZ,LL)=NFREQ(LZ,LL)*JCODE(LZ)
      LA=0
      DO 2572 LL=1,KTIME
      KZ=NTIME(LL)
      NFREQ(LZ,LL)=NFREQ(LZ,KZ)
      IF(NFREQ(LZ,LL).NE.0) LA=1
2572 CONTINUE
      IF(LA.EQ.0) GO TO 2564
      IF(FLAG(II))2566,2566,2563
2566 WRITE(LOUT,17) II, (NAMEM(L,LL),LL=1,4)
      NLINE=NLINE+1 $FLAG(II)=1
2568 IF(NLINE.GE.55) CALL HEADER(NLINE,NPAGE,I,LOUT,ITOPIC)
2567 WRITE(LOUT,33) (NAME(LZ,LP),LP=1,6),LZ, (NFREQ(LZ,LL),LL=1,KTIME)
      NLINE=NLINE+1
2564 CONTINUE
2570 CONTINUE
2575 CONTINUE
      CALL HEADER(NLINE,NPAGE,I,LOUT,ITOPIC)
2600 CONTINUE
      GO TO 6000

```

```

C -----
C THEME FREQUENCIES BY THEME
C -----

```

```

3999 DO 4500 I=1,NCOUN
      DO 4500 KK=1,MTIME
      BUFFER IN(LU,1) (MCOUNT(1),MCOUNT(NMAX))
4000 IF(UNIT,LU)4000,4001
4001 IF(NACON(I,3))4500,4500,4002
4002 DO 4400 J=1,NMAX
4400 NFREQ(J,KK)=NFREQ(J,KK)+MCOUNT(J)
4500 CONTINUE
4700 NPAGE=1
      WRITE(LOUT,11) (NCLASS(K),K=1,2),NPAGE, (NAME(J),J=1,KTIME)
      NLINE=7
      DO 4703 J=1,MTIME
      KZ=NTIME(J)
      IF(KZ.NE.0) NFREQ(I,J)=NFREQ(I,KZ)
4703 CONTINUE
      DO 4800 I=1,NMAX
      IF(JCODE(I))4800,4800,4702
4702 WRITE(LOUT,15) I, (NAME(I,J),J=1,6), (NFREQ(I,K),K=1,KTIME)
      NLINE=NLINE+1

```

```

      IF(NLINE.GE.55) CALL HEADER(NLINE,NPAGE,I,LOUT,ITOPIC)
4800  CONTINUE
      CALL HEADER(NLINE,NPAGE,1,LOUT,ITOPIC)
      GO TO 6000
C-----
C  COUNTRY PROFILE BY VARIABLE
C-----
3000  DO 3900 I=1,NCOUN
      DO 3001 L=1,MTIME
        BUFFER IN(LU,1)(NFREQ(1,L),NFREQ(NMAX,L))
3003  IF(UNIT,LU)3003,3001
3001  CONTINUE
        IF(NACON(I,3))3900,3900,3002
3002  NPAGE=NPAGE+1
        WRITE(LOUT,32)(NCLASS(K),K=1,2),NPAGE,(NACON(I,J),J=1,2),(NAMET(L)
1,L=1,KTIME)
        NLINE=7 $IFLAG=0
        DO 3200 J=1,NVAR
          IF(NTOPIC(J).EQ.0) GO TO 3200
          WRITE(LOUT,28)J,(NAMEM(J,LK),LK=1,4)
          NLINE=NLINE+3 $KZ=NAMEM(J,5) $IFLAG=IFLAG+1 $REWIND IFLAG
          READ(IFLAG) ((KTHEME(L,LK),LK=1,2),L=1,KZ)
          DO 3050 K=1,KZ
            LL=KTHEME(K,2)
            DO 3571 LK=1,MTIME
3571  NFREQ(LL,LK)=NFREQ(LL,LK)*JCODE(LL)
              L=0
              DO 3572 LJ=1,KTIME
                LZ=NTIME(LJ)
                NFREQ(LL,LJ)=NFREQ(LL,LZ)
                IF(NFREQ(LL,LJ).EQ.0) GO TO 3572
                MT(1)=I $MT(2)=J $MT(3)=LL $MT(4)=KTHEME(K,1) $MT(5)=LZ
                LXX=NFREQ(LL,LJ)
                DO 3573 LA=1,LXX
                  BUFFER OUT(LOUT,1)(MT(1),MT(5))
3570  IF(UNIT,LOUT)3570,3573
3573  CONTINUE
                  L=L+NFREQ(LL,LJ)
3572  CONTINUE
                  IF(L.EQ.0) GO TO 3050
3040  WRITE(LOUT,26) KTHEME(K,1),(NAME(LL,KK),KK=1,6),(NFREQ(LL,LZ),LZ=1
1,KTIME)
                  NLINE=NLINE+1
                  IF(NLINE.GE.55) CALL HEADER(NLINE,NPAGE,I,LOUT,ITOPIC)
3050  CONTINUE
                  REWIND IFLAG
3200  CONTINUE
                  CALL HEADER(NLINE,NPAGE,I,LOUT,ITOPIC)
3900  CONTINUE
      ENDFILE LOUT

```

```

GO TO 6000
1 FORMAT(9I2,4I1,2A8)
2 FORMAT(1X,* THE FOLLOWING COUNTRIES HAVE BEEN INCLUDED*,I5,* THRU*,I5)
  1,I5)
3 FORMAT(2I5)
4 FORMAT(1X,* ALL COUNTRIES WERE INCLUDED*)
5 FORMAT(1X,* THE FOLLOWING THEME NUMBERS WERE INCLUDED*,I5,* THRU*,
  1I5)
6 FORMAT(1X,* ALL THEME ARE INCLUDED*)
7 FORMAT(1X,*PROFILE TYPE*,I5,*CAN NOT BE USED WITH LTOPIC*,I5)
8 FORMAT(70I1)
9 FORMAT(1X)
11 FORMAT(1H1,4X,2A8,38X,*PAGE *,I3,///,7X,*THEME NO.*,10X,*THEME TIT
  1LE*,34X,*FREQUENCIES*,//,71X,4(A8,2X))
12 FORMAT(/,2X,I3,2X,6A8)
13 FORMAT(60X,2A8)
14 FORMAT(1H1,4X,2A8,38X,*PAGE *,I3,///,20X,*COUNTRY PROFILE FOR *,2A8,///,7X,
  18,///,7X,*THEME NO.*,10X,*THEME TITLE*,46X,*FREQUENCY*,//,71X,4(A8,2
  1X))
15 FORMAT(9X,I4,7X,6A8,5X,4(I3,7X))
16 FORMAT(1H1,4X,2A8,38X,*PAGE*,I4,///,20X,2A8,*(CON T)*,/)
17 FORMAT(6X,I2,2X,6A8)
18 FORMAT(1X,* ALL VARIABLES OR MAJOR/MINORS HAVE BEEN INCLUDED*)
19 FORMAT (5X,A8,A6)
20 FORMAT(16I5)
21 FORMAT(I5,6A8)
22 FORMAT(5X,4A8,18X,I5)
23 FORMAT(20I4)
25 FORMAT(10X,*.....CONTROL CARDS.....*,//,* LOGICAL UNIT OF INPUT.*,I
  15,/,* LOGICAL UNIT OF THE DICTIONARIES..*,I5,/,* LOGICAL UNIT OF T
  2HE OUTPUT..*,I5,/,* TYPE OF CCOUNTRY PROFILE..*,I5,/,* TYPE OF THEM
  3E PROFILE..*,I5,///,* SELECTION CRITERION OPERATING*,/)
26 FORMAT(10X,I5,2X,6A8,3X,4I8)
28 FORMAT(/,5X,I5,2X,4A8,/)
32 FORMAT(1H1,4X,2A8,38X,*PAGE *,I3,///,20X,*COUNTRY PROFILE FOR *,2A
  18,///,71X,4(A8,2X))
33 FORMAT(15X,6A8,*(*,I4,*).....*,4(I3,7X))
34 FORMAT(4A8)
6000 STOP
END

```

```

SUBROUTINE HEADER(NLINE,NPAGE,I,LOUT,ITOPIC)
COMMON NACON(160,3),NCLASS(2)

```

C
C
C
C

```

THIS SUBROUTINE PRODUCES THE TOP AND BOTTOM CLASSIFICATION LABELS AS
WELL AS CONTINUATION HEADERS AT THE TOP OF THE NEXT PAGE

```

```

IF(NLINE.LT.55) GO TO 100
NPAGE=NPAGE+1
WRITE(LOUT,13) (NCLASS(L),L=1,2)
IF(ITOPIC.EQ.1) WRITE(LOUT,11) (NCLASS(L),L=1,2),NPAGE
IF(ITOPIC.EQ.0) WRITE(LOUT,16) (NCLASS(L),L=1,2),NPAGE,(NACON(I,J),
1J=1,2)
NLINE=7
GO TO 150
100 DO 125 K=NLINE,55
125 WRITE(LOUT,9)
WRITE(LOUT,13) (NCLASS(L),L=1,2)
150 RETURN
9 FORMAT(1X)
11 FORMAT(1H1,4X,2A8,38X,*PAGE *,I3,///,7X,*THEME NO.*,10X,*THEME TIT
1LE*,34X,*FREQUENCIES*,//,71X,4(A8,2X))
13 FORMAT(60X,2A8)
16 FORMAT(1H1,4X,2A8,38X,*PAGE*,I4,///,20X,2A8,*(CON T)*,/)
END

```


SUMMARIZING COUNTRY POSITIONS

PROGRAM SCORES

Purpose

The primary purpose of this program is to calculate national scores (weighted averages of the remarks a state has made) for selected issue-variables. It allows the user to change any data on the issue-variable tape and to select the time periods and their weights for a specific analysis. Program SCORES must be run for each special analysis of up to 45 issue-variables.

The secondary purpose is to produce a tape of observation in a format that can be used by the BIMED34 regression package and to produce a tape of national scores for input into the MBLOC and VMODEL programs.

Method

The main problem card is read followed by the variable selection card that selects up to 45 issue-variables on which national scores are to be calculated (these issue-variables must exist on the issue-variable input tape). Then the time period weights are read. These operate for all issue-variables in this run. (See WEIGHT under dimensioned variables.) Finally, the themes that appear in more than one issue-variable being selected are read into MUSE. After these control cards are read, the dictionary files are read. The country names are read into NACON, the issue-variable titles into NAME, and the independent variables into NBLOC. Finally, the changes to the input data are read and NACON(N,3) is flagged with a 1 whenever there is a change to country N's data.

The observations are buffered in one at a time from the input tape. If the observation is on an issue-variable being processed, the program first determines whether it is a potential change. If it is, the list of changes is searched and the observation corrected in the manner specified. The way in which the observation is handled depends upon the weight of its time period if it is not a change and the frequency of mention if it is a change. By whichever method used, the observation is cumulated in VALUE, SIZE, MEAN, and SUM. If it is not a change and it refers to a multiply used theme, the observation may also be cumulated in NDOBL.

After all observations have been read off the input tape, any remaining changes are handled. The only changes that would remain are observations for countries that never addressed the issue-variables on the input tape. These are handled in the same manner as other changes.

The national score is then calculated for each country on each issue-variable. Simultaneously the mean and standard deviation are calculated for each issue-variable. A BCD regression tape and a binary national scores tape are written if they have been requested on the main problem card. Finally, the national scores are printed by country and/or issue-variable depending upon the user's request.

Structure of the Output Tapes

Two tapes are produced by this program - a BCD regression tape and a binary national scores tape. The regression tape is written in a format acceptable to BIMED34. Seventy files are written, one for each variable; but as a restriction in program SCORES only 45 variables could have data. Each file contains a different number of records; the number depends upon the number of countries with national scores for this issue-variable. The records are organized by country number. For each country with data, a record containing 36 entries is written. The first entry is its national score, the second through 35th are its discrete independent variables (see the users guide for a description of the 6th file of the dictionary, particularly its first 34 entries), and the 36th entry is the number of remarks by the country used to calculate its national score. This tape is written in the format F6.2,34F3.1,F5.1.

The national scores tape has two files. The first file consists of the national scores of all countries on all the issue-variables followed by the frequencies of mention. The issue-variables appear in numerical order with two records each. The first record contains the national scores of all countries; the second, the frequencies. The second file consists of one record which contains the number of times each country has mentioned a multiply used theme. The countries' frequencies on the first theme appear before the frequencies on the second, etc.

Dimensioned Variables

There are 16 dimensioned variables in program SCORES:

(1) VALUE is a matrix dimensioned 149 by 45. A row represents a country and a column one of the issue-variables being processed. It contains the national score of a country on an issue-variable. As the observations are being cumulated, VALUE(N,M) contains the sum of the ranks of themes country N has mentioned on the Mth issue-variable being processed. This is later divided by the number of observations by this country (SIZE) to obtain a national score.

(2) SIZE is a matrix dimensioned 149 by 45. A row is a country and a column is one of the issue-variables being processed. It contains the number of observations a country has made on each of the issue-variables.

(3) MEAN, dimensioned 45, contains the mean of each of the 45 possible issue-variables. As the observations are being read, MEAN(M) contains the sum of all ranks of themes mentioned by countries on the Mth issue-variable being processed. This is later divided by the total number of observations (SUM) to obtain a mean.

(4) STD, dimensioned 45, is used to calculate the standard deviation of each of the 45 possible issue-variables. As the observations are read, STD(M) contains the squared sum of the ranks on the Mth issue-variable being processed. This is later used to calculate the standard deviation.

(5) NVAR, dimensioned 70, contains a flag for selecting up to 45 of the 70 issue-variables. If NVAR(M) is 1, variable M is selected; if it is 0, M is not selected.

(6) NAME, dimensioned 70 by 4, is a matrix of issue-variable titles. For each issue-variable, a four field title is stored.

(7) NBLOC is dimensioned 149 by 35. A row represents a country and a column represents one of the discrete independent variables. For a discussion of these variables, see appendix B of the users guide, and particularly note the first 35 variables in file 6.

(8) NCHANGE, dimensioned 300 by 4, is a matrix of changes to the data. It allows up to 300 changes to be read in. Each change has four entries: (1) country number, (2) issue-variable number, (3) rank within issue-variable, and (4) number of mentions. After a change has been processed, the fourth field is set to -9 as a flag that it has been completed.

(9) NACON, dimensioned 149 by 3, is a matrix of country names and country change flags. For each country, the first two fields contain its name and the third field is a 1 if at least one change will be made to the country's data and a 0 otherwise.

(10) WEIGHT, dimensioned 4, represents the weight to be assigned the four possible time periods on the input tape. If WEIGHT(J) is 0, all data from time period J is excluded from this analysis. If WEIGHT(J) is 1, all data from time period J is included. If WEIGHT(J) is 2 (e.g.), all data will be included, but it will be entered twice (e.g.). Any positive integer or zero weight can be assigned.

(11) SUM, dimensioned 45, represents the total number of observations on each of the 45 possible issue-variables being processed.

(12) MUSE, dimensioned 10 by 2, contains up to 10 multiply used themes. For each theme, the number is followed by one of the issue-variables being processed that this theme is in.

(13) NDOBL is dimensioned 149 by 10. A row represents a country and a column represents one of the 10 possible multiply used themes. The matrix contains the number of times a country has said one of these themes.

(14) JCODE, dimensioned 1600, serves two purposes. First, it is temporary storage of the themes within issue-variables and their ranks. Second, it flags the themes that are used more than once. JCODE(M) is 1 when theme M is a multiply used theme and 0 otherwise.

(15) MCOUNT, dimensioned 5, is the temporary storage of the five components of an observation that is buffered in from the input tape. The five entries are (1) country number, (2) issue-variable number, (3) theme number, (4) rank of the theme within this issue-variable, and (5) time period in which this remark was made (1-4).

(16) BLANK, dimensioned 149, contains a zero in each field. It is used to fill the national scores output tape with a zero for each issue-variable not processed.

Operational Effects of System Changes

This program depends upon the issue-variable tape (produced by PROFLE) and four dictionary files - country, theme, issue-variable and independent variable files. Assuming that their structure and format remain the same, the following changes to the dictionary files can be handled by changing these variables:

If the number of countries exceeds 149, VALUE, SIZE, NBLOC, NACON, and NDOBL must be redimensioned.

If the number of issue-variables exceeds 70, LMAX and KVAR must be increased and NVAR and NAME must be redimensioned.

If the maximum theme number exceeds 1600, MAXTHM will have to be increased and JCODE redimensioned.

If the number of independent variables exceeds 34 or others are to be used, MAXB may have to be increased, NBLOC redimensioned, and formats 6 and 38 changed.

Apart from dictionary file changes, it is possible that in the future the input tape will contain more than 4 time periods. If this occurs, MTIME will have to be increased, WEIGHT redimensioned, and format 1 perhaps changed. An increase in time periods means that programs MATRIX and PROFLE would have been expanded as well.

In addition, there are internal changes that do not affect other programs. If more than 300 changes are required, NCHANGE can be redimensioned without affecting other programs. If the number of multiply used themes exceeds 10, MUSE and NDOBL will have to be redimensioned and program MAXPAC changed slightly.

Program Listing

The listing for program SCORES follows.

PROGRAM SCORES

COMMON VALUE (149,45),SIZE (149,45),MEAN (45),STD (45),BLANK (149)
 DIMENSION NVAR (70),NAME (70,4),NBLOC (149,35),NCHANGE (300,4)
 DIMENSION NACON (149,3),WEIGHT (4),SUM (45),MUSE (10,2)
 DIMENSION NDOBL (149,10),JCODE (1600),MCOUNT (5)
 TYPE REAL NBLOC,MEAN,NDOBL
 TYPE INTEGER WEIGHT,SUM

C
C
C
C

INITIALIZE VARIABLES...READ MAIN PROBLEM CARD,VARIABLE SELECTION
 CARD, WEIGHTS AND MULTIPLY USED CARDS

MAXB=35 \$MAXTHM=1600 \$MTIME=4 \$LMAX=70 \$KVAR=70
 READ 5,NCOUN,LCUTT,LIN,ITAPE,LDICT,NDOUB,LCHANGE,LREG,LTAPE,JVAR,J
 1COUN
 READ 3,(NVAR(I),I=1,KVAR)
 READ 1,(WEIGHT(I),I=1,MTIME)
 PRINT 1,(WEIGHT(I),I=1,MTIME)
 IF (NDOUB.NE.0) READ 5,((MUSE(I,J),J=1,2),I=1,NDOUB)

C
C
C

READ COUNTRY, VARIABLE AND INDEPENDENT VARIABLE DICTIONARIES

DO 27,I=1,NCCUN
 NACON(I,3)=0 \$BLANK(I)=0.
 27 READ(LDICT,13)(NACON(I,J),J=1,2)
 DO 39 M=1,3
 39 CALL SKIPFILE (LDICT)
 N=1
 DO 50 I=1,KVAR
 READ(LDICT,4)(NAME(I,J),J=1,4),NTHEME
 NTHEME=NTHEME*2
 READ(LDICT,15)(JCODE(M),M=1,NTHEME)
 15 FORMAT(8(I5,I5))
 IF (NVAR(I))50,50,41
 41 NVAR(I)=N
 N=N+1
 50 CONTINUE
 KVAR=N-1
 DO 40 M=1,2
 40 CALL SKIPFILE (LDICT)
 DO 37 K=1,NCOUN
 37 READ(LDICT,38)((NBLOC(K,M),M=2,MAXB))
 DO 51 I=1,MAXTHM
 51 JCODE(I)=0
 DO 52 I=1,NDOUB
 M=MUSE(I,1)
 52 JCODE(M)=I
 DO 59 I=1,NCOUN
 DO 48 J=1,10
 48 NDOBL(I,J)=0
 DO 59 J=1,KVAR

```

      MEAN(J)=0.  $VALUE(I,J)=0.  $SIZE(I,J)=0.  $SUM=0
59  STD(J)=0.
C
C      READ CHANGES TO THE INPUT TAPE
C
      DO 55 I=1,LCHANGE
      READ 2,(NCHANGE(I,K),K=1,4)
      PRINT 2,(NCHANGE(I,K),K=1,4)
      J=NCHANGE(I,1)
55  NACON (J,3) = 1
C
C      READ AN OBSERVATION AND CHECK WHETHER IT BELONGS TO A VARIABLE
C      BEING PROCESSED...IF IT IS,IS IT ONE THAT IS CHANGING
C
      60  BUFFER IN(LIN,1) (MCCOUNT(1),MCCOUNT(5))
      62  IF(UNIT,LIN) 62,61,100
      61  ICC=MCCOUNT(1)  $IVAR=MCCOUNT(2)  $ITHEME=MCCOUNT(3)
      IRANK=MCCOUNT(4)  $ITIME=MCCOUNT(5)
      K=NVAR(IVAR)
      IF(NVAR(IVAR)) 60,60,66
      66  KFLAG=0
      ITHE=0
      IF(NACON(ICC,3).NE.0) 63,69
C
C      YES IT IS CHANGING..SEARCH LIST OF CHANGES AND SUBSTITUTE CHANGE
C
      63  DO 64 I=1,LCHANGE
      IF(ICC.NE.NCHANGE(I,1)) GO TO 64
      IF(IVAR.NE.NCHANGE(I,2)) GO TO 64
      IRANK=NCHANGE(I,3)  $ITHE=I  $KFLAG=NCHANGE(I,4)
      64  CONTINUE
C
C      DETERMINE WEIGHT TO BE ASSIGNED THIS OBSERVATION AND FILL THE MATRIX
C      BY WEIGHTING BY THE EMPHASIS CRITERION FOR THIS TIME PERIOD
C
      69  LX=WEIGHT(ITIME)
      IF(KFLAG) 60,80,70
      70  NCHANGE(ITHE,4)=-9
      LX=KFLAG
      80  DO 67 I=1,LX
      VALUE(ICC,K)=VALUE(ICC,K)+FLOAT(IRANK)
      SIZE(ICC,K)=SIZE(ICC,K)+1.
      MEAN(K)=MEAN(K)+FLOAT(IRANK)
      STD(K)=STD(K)+(FLOAT(IRANK)*FLOAT(IRANK))
      SUM(K)=SUM(K)+1
      IF(LX.EQ.KFLAG) GO TO 67
      IF(JCODE(ITHEME).EQ.0) GO TO 67
      M=JCODE(ITHEME)  $MM=MUSE(M,2)
      IF(IVAR.EQ.MM) NDOBL(ICC,M)=NDOBL(ICC,M)+1.
      67  CONTINUE

```

```

      GO TO 60
C
C   AFTER READING ALL OBSERVATION ARE THERE ANY CHANGES LEFT TO DO
C
100 DO 90 I=1,LCHANGE
    IF (NCHANGE(I,4)) 90,90,81
    81 LX=NCHANGE (I,4)
      IVAR=NCHANGE (I,2)  $IRANK=NCHANGE(I,3)  $ICC=NCHANGE(I,1)
      K=NVAR(IVAR)
      DO 82 J=1,LX
        VALUE(ICC,K)=VALUE(ICC,K)+FLOAT(IRANK)
        SIZE(ICC,K)=SIZE(ICC,K)+1.
        MEAN(K)=MEAN(K)+FLOAT(IRANK)
        STD(K)=STD(K)+(FLOAT(IRANK)*FLOAT(IRANK))
    82 SUM(K)=SUM(K)+1
    90 CONTINUE
C
C   CALCULATE NATIONAL SCORES FOR EACH VARIABLE BY COUNTRY...
C   CALCULATE THE MEAN AND STANDARD DEVIATION AND SAMPLE SIZE...PRINT
C
    DO 110 J=1,KVAR
    DO 109 I=1,NCCUN
      IF (SIZE(I,J).NE.0.) VALUE(I,J)=VALUE(I,J)/SIZE(I,J)
    109 CONTINUE
    110 CONTINUE
      DO 120 J=1,KVAR
        IF (SUM(J).EQ.0) GO TO 120
        STD(J)=SQRT((STD(J)-((MEAN(J)*MEAN(J))/SUM(J)))/(SUM(J)-1))
        MEAN(J)=MEAN(J)/SUM(J)
    120 CONTINUE
      LQ=0
      PRINT 131
      DO 135 J=1,LMAX
        IF (NVAR(J).EQ.0) GO TO 135
        LQ=LQ+1
        PRINT 132,J,MEAN(LQ),STD(LQ),SUM(LQ)
    135 CONTINUE
C
C   WRITE THE REGRESSION INPUT TAPE..IF REQUESTED
C   THEN WRITE THE NATIONAL SCORES TAPE IF REQUESTED
C
      IF (LREG.EQ.0) GO TO 200
      DO 150 I=1,LMAX
        K=NVAR(I)  $LQ=0
        IF (NVAR(I).EQ.0) 150,142
    142 DO 143 J=1,NCCUN
        IF (VALUE(J,K).EQ.0.) GO TO 143
        LQ=LQ+1
        WRITE(LOUTT,6) VALUE(J,K), (NBLOC(J,JK), JK=2,MAX0), SIZE(J,K)
    143 CONTINUE

```

```

      PRINT 5,I,LQ
150 ENDFILE LOUIT
200 IF(LTAPE.EQ.0) GO TO 300
      DO 225 J=1,LMAX
      KK=NVAR(J)
      IF(KK.LE.0) 201,219
201 BUFFER OUT(ITAPE,1)(BLANK(1),BLANK(NCOUN))
202 IF(UNIT,ITAPE) 202,203
203 BUFFER OUT(ITAPE,1)(BLANK(1),BLANK(NCOUN))
204 IF(UNIT,ITAPE) 204,225
219 BUFFER OUT(ITAPE,1)(VALUE(1,KK),VALUE(NCOUN,KK))
220 IF(UNIT,ITAPE) 220,221
221 BUFFER OUT(ITAPE,1)(SIZE(1,KK),SIZE(NCOUN,KK))
222 IF(UNIT,ITAPE) 222,225
225 CONTINUE
      ENDFILE ITAPE
      IF(NDOUB.LE.0) GO TO 227
      BUFFER OUT(ITAPE,1)(NDOEL(1,1),NDOBL(NCOUN,NDOUB))
226 IF(UNIT,ITAPE) 226,227
227 ENDFILE ITAPE

C
C
C      PRINT NATIONAL SCORES BY COUNTRY...IF REQUESTED
300 IF(JCOUN.EQ.0) GO TO 400
      REWIND LIN
      DO 350 I=1,NCCUN
      PRINT 9,(NACON(I,J),J=1,2)
      DO 340 J=1,LMAX
      IF(NVAR(J)) 340,340,301
301 LQ=NVAR(J)
      PRINT 10,J,(NAME(J,KZ),KZ=1,4),VALUE(I,LQ),SIZE(I,LQ)
340 CONTINUE
350 CONTINUE

C
C
C      PRINT NATIONAL SCORES BY ISSUE-VARIABLE...IF REQUESTED
400 IF(JVAR.EQ.0) GO TO 500
      DO 450 J=1,LMAX
      LQ=NVAR(J)
      IF(NVAR(J)) 450,450,401
401 PRINT 11,J,(NAME(J,KZ),KZ=1,4)
      DO 440 I=1,NCCUN
      IF(VALUE(I,LQ).NE.0.) PRINT 12,(NACON(I,K),K=1,2),VALUE(I,LQ),SIZE(I,LQ)
      1(I,LQ)
440 CONTINUE
450 CONTINUE

C
C
C      FORMAT STATEMENTS
1 FORMAT(7I2)

```



```

2 FORMAT (5I5)
3 FORMAT (80I1)
4 FORMAT (5X,4A8,18X,I5)
5 FORMAT (16I5)
6 FORMAT (F6.2,34F3.1,F5.1)
7 FORMAT (20F6.2)
8 FORMAT (45F5.2)
9 FORMAT (/////10X,*PROFILE FOR *,A8,A7,///,38X,*NATIONAL      NO. OF*
1,/,39X,*SCORES      OBSERVATIONS*,/)
10 FORMAT (2X,I2,X,4A8,2(F10.3,3X))
11 FORMAT (* VARIABLE*,I2,* *,4A8,/)
12 FORMAT (5X,A8,A6,F10.3,F7.0)
13 FORMAT (5X,A8,A6)
38 FORMAT (4X,34F3.1,////////)
131 FORMAT (1H1,* VARIABLE      MEAN      STD      SIZE *,////)
132 FORMAT (5X,I5,2F10.2,I10)
500 END

```

PROGRAM BIMED34

BIMED34¹ is a standard statistical program available at CNA which estimates a multiple linear regression equation in a stepwise manner. BIMED34 was selected because it provides numerous features that are necessary for an LOS issue analysis. It allows regression problems on different data bases to be stacked back to back; it can compute a weighted regression; it can force the regression equation through the origin; and it produces machine-readable output.

The version of BIMED34 used by the LOS project differs from the CNA version in the way in which the independent variable numbers are punched on the output cards. The CNA version assigns variable numbers 1-N to its N output cards. The LOS version assigns the original variable number. That is, if independent variable 3 is in the equation, the LOS version will punch a 3 as its variable number, whereas the CNA version will punch a K when variable 3 is the Kth variable in the equation.

Since MBLOC and VMODEL will read these output cards and match the variable numbers punched with the corresponding independent variable data, the LOS version is essential to normal operations of these two programs. Any modifications to BIMED34 should be made to the LOS version.

If a listing of BIMED34 is requested, a CNA systems programmer should be consulted. The program is on a load and go tape.

PROGRAM MBLOC

Purpose

The functions of this program are to calculate regression estimates and preferred positions for all countries, to calculate mean preferred positions and standard deviations for the 34 blocs or groups, and to produce a series of summary reports of group positions on selected issue-variables. As many as 45 variables can be processed in a single run.

Method

After the main problem and issue-variable selection cards are read, the deck of beta coefficients is read and placed in BETA. Then the names of blocs are read. Finally, the country and independent variable files are read off the dictionary tape.

¹Hodson Thornber, "Manual for (B34T, 8 Mar 66) A Stepwise Regression Program," Technical Report 6603, Center for Mathematical Studies in Business and Economics, University of Chicago. Copies are available from the Information Document Center of CNA.

For a selected issue-variable, the national scores and frequencies of mention for all countries are buffered in from the input tape and placed in MATRIX(country no., 1 and 4). Then a regression estimate and a preferred position are calculated for each country and placed in MATRIX(country no., 2 and 3).

For each bloc, the data from all its members is cumulated in MBLOK and a mean and a standard deviation are calculated from their national scores alone, estimates alone and preferred positions. These statistics are then printed in a summary table in which the blocs appear in increasing order by mean preferred position. At this point in the program, a more detailed report by bloc can be produced; if requested, the members of each bloc are listed in order of their preferred positions. Then the program returns to process another selected issue-variable.

Dimensioned Variables

There are eight dimensioned variables in program MBLOC:

(1) KVAR, an array dimensioned 45, contains the numbers of the issue-variables being processed. The variables are listed in numerical order.

(2) BETA is a matrix dimensioned 45 by 34. Each row represents an issue-variable being processed; each column represents a discrete independent variable (or bloc affiliation). The matrix contains the beta coefficient from BIMED34 for the issue-variable and bloc.

(3) BNAME, an array dimensioned 34, contains a one field name for each of the independent variables or blocs.

(4) NACON, dimensioned 149 by 2, is a matrix of country names. Each name is assigned two fields.

(5) BLOC is a matrix 149 by 34. A row represents a country and a column an independent variable or bloc. This matrix contains a 1 if the country is a member of the bloc and a 0 otherwise.

(6) MATRIX, dimensioned 149 by 4, contains all the data on the issue-variable being processed. There are four entries for each country: (1) national score, (2) regression estimate, (3) preferred position, and (4) frequency of mentions.

(7) MBLOK, a three dimensional matrix 34 by 4 by 2, contains all the aggregate statistics of the 34 blocs on the issue-variable being processed. For each bloc J, MBLOK is as follows:

(J, 1, 1) = mean of the national scores of its members

(J, 1, 2) = standard deviation of the national scores of its members

(J, 2, 1) = mean of the regression estimates of its members

(J, 2, 2) = standard deviation of the regression estimates of its members

(J, 3, 1) = mean of the preferred positions of its members

(J, 3, 2) = standard deviation of the preferred positions of its members
(J, 4, 1) = number of members with national scores
(J, 4, 2) = total number of members.

(8) NORDER, dimensioned 149, serves as a flag when blocs or countries are reordered. If NORDER(K) is 1, bloc or country K has been written on the printout. If it is 0, bloc or country K will be printed whenever it becomes the next in the order list.

Operational Effects of System Changes

This program depends upon the national scores tape from program SCORES and the country and independent variable files of the dictionary tape. Assuming that their structure and format remain the same, the following modifications can be handled in this program by changing these variables:

If the number of countries exceeds 149, BLOC, MBLOC, MATRIX, NACON, and NORDER will have to be redimensioned.

If the number of discrete independent variables exceeds 34, BLOC, MBLOC, BETA, and BNAME will have to be redimensioned and NB increased. Furthermore, format 11, which reads independent variables off the dictionary tape, will have to be changed. (Note: If format is changed, SCORES and BIMED34 must be changed accordingly.)

It is possible that in the future the regression analysis may involve continuous as well as discrete variables. If so, MBLOC must be changed substantially followed by corresponding changes in SCORES, BIMED34, and VMODEL.

Program Listing

The listing of program MBLOC follows.

```

PROGRAM MBLOC
DIMENSION BLOC(149,34),MBLOK(34,4,2),MATRIX(149,4),NORDER(149)
DIMENSION KVAR(45),BETA(45,34),BNAME(34),NACON(149,2)
TYPE REAL MATRIX,MBLOK
TYPE INTEGER SUMM

C
C READ MAIN PROBLEM CARD, VARIABLE SELECTION CARD,BETAS,NAMES OF BLOCs
C THEN READ THE COUNTRY AND INDEPENDENT VARIABLES FILES...

READ 8,LDICT,LIN,NVAR,NCOUN,SUMM
READ 1,(KVAR(J),J=1,NVAR)
NB=34 $KCOUNT=0
DO 40 I=1,NVAR
DO 30 M=1,NB
30 BETA(I,M)=0.
READ 1,NBETA
DO 35 M=1,NBETA
READ 9,NUMB,VBETA
NUMB=NUMB-1
35 BETA(I,NUMB)=VBETA
40 PRINT 10,((M,BETA(I,M)),M=1,NB)
READ 3,(BNAME(I),I=1,NB)
READ(LDICT,4)((NACON(I,J),J=1,2),I=1,NCOUN)
DO 41 M=1,5
41 CALL SKIPFILE(LDICT)
READ(LDICT,11)((BLOC(I,K),K=1,NB),I=1,NCOUN)

C
C READ NATIONAL SCORES AND SALIENCES FOR ALL COUNTRIES
C ON THE VARIABLE BEING PROCESSED

INDEX=0
DO 100 II=1,NVAR
31 DO 25 I=1,NCOUN
DO 25 J=1,4
25 MATRIX(I,J)=0.
BUFFER IN(LIN,1)(MATRIX(1,1),MATRIX(NCOUN,1))
26 IF(UNIT,LIN)26,27
27 BUFFER IN(LIN,1)(MATRIX(1,4),MATRIX(NCOUN,4))
28 IF(UNIT,LIN)28,29
29 INDEX=INDEX+1
IF(INDEX.NE.KVAR(II)) GO TO 31

C
C CALCULATE THE REGRESSION ESTIMATE AND PREFERRED POSITION FOR EACH
C COUNTRY ON THE VARIABLE BEING PROCESSED...

DO 211 I=1,NCOUN
DO 200 K=1,NB
200 MATRIX(I,2)=MATRIX(I,2)+(BLOC(I,K)*BETA(II,K))
211 MATRIX(I,3)=((MATRIX(I,1)*MATRIX(I,4))+MATRIX(I,2))/(MATRIX(I,4)+1
1.)

```

```

C
C
C
C
C
CUMULATE THE DATA FOR EACH BLOC IN MBLOK AND CALCULATE THE MEAN
AND STANDARD DEVIATION OF EACH BLOCS DATA FOR NATIONAL SCORES ONLY,
REGRESSION ESTIMATES ONLY AND PREFERRED POSITIONS...
C
DO 600 I=1,NB
DO 600 J=1,4
DO 600 K=1,2
600 MBLOK(I,J,K)=0.
DO 400 I=1,NCCUN
DO 400 J=1,NB
IF(BLOC(I,J).EQ.0.) GO TO 400
IF(MATRIX(I,1).NE.0.) MBLOK(J,4,1)=MBLOK(J,4,1)+1.
MBLOK(J,4,2)=MBLOK(J,4,2)+1.
DO 410 K=1,3
MBLOK(J,K,1)=MBLOK(J,K,1)+MATRIX(I,K)
410 MBLOK(J,K,2)=MBLOK(J,K,2)+(MATRIX(I,K)*MATRIX(I,K))
400 CONTINUE
DO 500 J=1,NB
DO 500 K=1,3
OBS=MBLOK(J,4,2)
IF(K.EQ.1) OBS=MBLOK(J,4,1)
MBLOK(J,K,2)=SQRT((MBLOK(J,K,2)-(MBLOK(J,K,1)*MBLOK(J,K,1))/OBS)/(
1OBS-1.))
500 MBLOK(J,K,1)=MBLOK(J,K,1)/OBS
C
C
C
C
PRINT SUMMARY TABLE OF BLOCS IN INCREASING ORDER BY THE MEAN OF
THEIR PREFERRED POSITIONS....
C
PRINT 6,KVAR(II)
DO 501 KL=1,NB
501 NORER(KL)=0
DO 502 K=1,NB
OBS=999.
DO 503 L=1,NB
IF(MBLOK(L,3,1).LE.OBS.AND.NORDER(L).EQ.0) 504,503
504 OBS=MBLOK(L,3,1)
NCOUNT=L
503 CONTINUE
PRINT 7,BNAME(NCOUNT),((MBLOK(NCOUNT,KK,LL),LL=1,2),KK=1,4)
NORDER(NCOUNT)=1
502 CONTINUE
C
C
C
C
IF REQUESTED, PRINT FOR EACH BLOC A LIST OF IT MEMBERS AND THEIR
PREFERRED POSITIONS IN INCREASING ORDER...
C
IF(SUMM.EQ.1) GO TO 100
DO 1900 JJ=1,NB
PRINT 1801,KVAR(II),BNAME(JJ),MBLOK(JJ,3,1)
DO 1800 I=1,NCCUN

```

```

1800 NORDER(I)=0
      NOBS=MBLOC(JJ,4,2)
      NCOUNT=0
      DO 1802 I=1,NCOUN
      IF(BLOC(I,JJ).EQ.0.) 1802,1803
1803 NCOUNT=NCOUNT+1
      IF(NCUNT.EQ.1)1804,1805
1804 NORDER(NCOUNT)=I
      GO TO 1802
1805 KZZ=NCOUNT-1
      KZ=0
      DO 1850 JK=1,KZZ
      L=NORDER(JK)
      IF(KZ.NE.0) GO TO 1850
      IF(MATRIX(I,3).LT.MATRIX(L,3))KZ=JK
1850 CONTINUE
      IF(KZ.EQ.0) GO TO 1804
      LL=NCOUNT
      DO 1807 KK=KZ,KZZ
      NORDER(LL)=NORDER(LL-1)
1807 LL=LL-1
      NORDER(KZ)=I
1802 CONTINUE
      DO 1860 JK=1,NOBS
      L=NORDER(JK)
      PRINT 1861,(NACON(L,LL),LL=1,2),MATRIX(L,1),MATRIX(L,2),MATRIX(L,3
1)
1860 CONTINUE
1900 CONTINUE
100 CONTINUE

```

C
C
C

FORMAT STATEMENTS

```

1 FORMAT(40I2)
2 FORMAT(8X,E12.4)
3 FORMAT(10A8)
4 FORMAT(5X,A8,A6)
6 FORMAT(1H1,* THE FOLLOWING MEANS AND STANDARD DEVIATIONS BY BLOCs
1 ARE FOR VARIABLE *,I4,///,1X,*NAME OF DATA DATA EST
2 EST SCORE SCORE DATA UN*,/,3X,*BLOC MEAN STD
3 MEAN STD MEAN STD SS SS*,//)
7 FORMAT(1X,A8,3X,6(F6.2,2X),2F6.0)
8 FORMAT(5I5)
9 FORMAT(6X,I2,E12.4)
10 FORMAT(9(I3,F10.1))
11 FORMAT(4X,34F3.1,////////)
1801 FORMAT(1H1,3X,*SCORES OF COUNTRIES ON VARIABLE*,I3,///3X,*THAT BELO
ING TO THE BLOC*,A8,///,10X,*BLOC MEAN IS*,F6.2,///,3X,*COUNTRY D
2ATA ESTIMATE*,//)
1861 FORMAT(1X,A8,A3,3F10.3)
      REWIND LIN
      END

```

FORECASTING OUTCOMES

PROGRAM VMODEL

Purpose

This program forecasts outcomes on selected individual issue-variables. For each issue-variable, medians, best and worst cases, and the preferred choice within a pair of alternatives can be produced. In addition, the program produces a tape that is used by program MAXPAC to analyze packages of issues.

Method

After the main problem and issue-variable selection cards are read, the country and independent variable files are read off the dictionary tape. For each of the selected issue-variables, the beta coefficient deck and input records containing the national scores and frequencies of mention for all countries are read. These enable the program to calculate a regression estimate, a residual (the difference between a country's national score, if it exists, and its estimate), and a preferred position for each country. If the estimate exceeds the bounds of the issue-variable, the estimate is set equal to the upper or lower bound. The output table contains all of these statistics and medians of the national scores, estimates and preferred positions. The program returns to process the next selected issue-variable. The program then turns to best and worst cases. For each case, the program calculates the number of countries with preferred positions within the range specified. It lists the countries with their preferred positions and a tally of the number of states.

Then the matrix ISSUE, which contains the preferred position and frequency of mentions for each country on each of the selected issue-variables, is standardized by subtracting the mean of the issue-variable from each of the preferred positions and dividing this difference by the standard deviation of the issue-variable. This is printed and then written on the output tape in binary. From this point on in the program, ISSUE represents the normalized or standardized preferred positions.

The last section of this program calculates the outcome on an issue when countries are forced to choose between two alternatives. For each such forecast, the two alternatives are read and then standardized in the manner described above. A utility equation is used to determine which of the two alternatives each country prefers. After utility scores have been calculated for all countries, a summary outcome table is produced and, if requested, a list of countries with their utility scores is printed. This list of countries is ordered by intensity of preference. The program returns to process another pair of alternatives on this or another issue-variable.

For a discussion of the formulas used throughout this program and the way in which the results are interpreted, the programmer should refer to the users guide.

Structure of the Output Tape

The output tape from this program is used as input to MAXPAC, which analyzes packages of issues, and VALID, which compares the forecasts with 'outside' forecasts. It is written in binary and contains four kinds of records. The first record, 70 fields filled with 0's or counters, contains a list of the issue-variables selected for this run. The second record contains the number of issue-variables and countries processed in this program that will appear on this output tape. The third record contains a mean and a standard deviation for each of the selected issue-variables. The fourth series of records contains the matrix ISSUE (which is seen under Dimensioned Variables).

Dimensioned Variables

There are 12 dimensioned variables in program VMODEL:

(1) BLOC, a matrix dimensioned 149 by 35, contains the country number followed by the 34 discrete independent variables for each of the 149 countries. For a definition of these 34 variables, see the discussion of dictionary file 6 in the users guide.

(2) BETA, an array dimensioned 35, contains the beta coefficients for the issue-variable being processed.

(3) MEDIAN, an array dimensioned 3, contains the median national score, median estimate and median preferred position for the issue-variable being processed.

(4) NOBS, an array dimensioned 45, contains the number of countries with national scores on each of the 45 possible issue-variables that can be processed in a single run.

(5) KVAR has 70 fields, one for each existing issue-variable number. It contains flags for selecting issue-variables for this run. Originally it is read as a series of 0's and 1's, where a 1 means that the issue-variable is selected and a 0 that it is not. However, the program converts all the 1's to counters so that $KVAR(N) = J$ when issue-variable N is the Jth one selected for processing.

(6) NACON, dimensioned 149 by 2, is a matrix of country names. Each country is assigned a two field name from the country dictionary.

(7) MEAN, dimensioned 45, is used in the calculation and ultimately the storage of the means of the 45 possible issue-variables.

(8) STD, dimensioned 45, is used in the calculation and ultimately the storage of the standard deviations of the 45 possible issue-variables.

(9) VOTE, an array dimensioned 3, contains the number of states that prefer alternative 1, that are indifferent, and that prefer alternative 2.

(10) CURMIN, an array dimensioned 3, is used in the calculation of the median national score, median estimate and median preferred position. In each case, CURMIN represents the current minimum score found in the list.

(11) SCORE, an array dimensioned 149, contains the scores for all 149 countries that were stored on the disc. The scores could be national scores, estimates, or preferred positions, depending upon the disc from which they were taken.

(12) ISSUE, a three dimensional matrix 149 by 2 by 45, contains the standardized preferred position and frequency of mentions of each country on each of the 45 possible issue-variables being processed. These two statistics appear as a set for all countries on one issue-variable at a time.

Operational Effects of System Changes

Program VMODEL depends upon the national scores tape and the country and independent variable files of the dictionary tape. Assuming that their structure and format remain the same, the following minor changes to the dictionary files can be handled by changing the following variables:

If the number of countries exceeds 149, ISSUE, BLOC, NACON, and SCORE will have to be redimensioned.

If the number of independent variables used in the regression analysis exceeds 34 or the variables in the regression are not the 34 discrete independent variables listed in the 2nd through the 35th position in the sixth file of the dictionary, format 4 must be changed. Furthermore, if more than 34 variables were processed through BIMED34, MAXB must be set equal to the number of variables plus 1 and BLOC and BETA must also be dimensioned one more than the number of variables processed.

If the number of issue-variables ever exceeds 70, MAXV must be increased and KVAR redimensioned.

Apart from these changes to the dictionary files, it may become desirable in the future to process more than 45 issue-variables in a single run. If this is to be done, ISSUE, NOBS, MEAN, and STD would have to be redimensioned and MAXI increased. The programmer is cautioned that expanding the number of issue-variables would serve little purpose without a corresponding change in programs PROFLE, SCORES, MBLOC, and MAXPAC.

Program Listing

The listing of program VMODEL follows.

AD-A132 589

FORECASTING OUTCOMES OF MULTILATERAL NEGOTIATIONS:
COMPUTER PROGRAMS VOLU. (U) CENTER FOR NAVAL ANALYSES
ALEXANDRIA VA INST OF NAVAL STUDIES. K W GOUDREAU

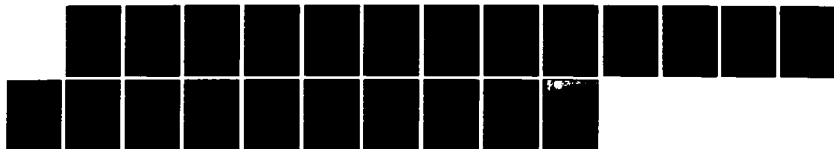
272

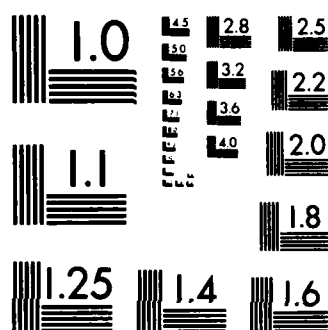
UNCLASSIFIED

OCT 76 CRC-290-VOL-2 N00014-76-C-0001

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```

PROGRAM VMODEL
DIMENSION ISSUE(149,2,45),BLOC(149,35),BETA(35),MEDIAN(3)
COMMON MEAN(45),STD(45),NVAR,NCOUN
DIMENSION NOBS(45),KVAR(70),NACON(149,2)
DIMENSION VOTE(3),CURMIN(3),SCOKE(149)
TYPE REAL MIN,MAX,MEAN,ISSUE,MRESID,MEDIAN
TYPE INTEGER BLANK,VOTE

```

```

C
C INITIALIZE MAXIMUM VALUES...READ MAIN PROBLEM CARD, VARIABLE
C SELECTION CARD, AND THE DICTIONARIES
C

```

```

MAXB=35 $MAXV=70 $MAXI=45
DO 90 M=1,MAXI
90 NOBS(M) = 0
READ 1,NVAR,NRANGE,NSINGLE,LDICT,LUIN,NOMED,NOLIST,NCOUN
READ 2, (KVAR(M),M=1,MAXV)
INUM = 0
DO 95 M=1,MAXV
IF (KVAR(M) .LE. 0) GO TO 95
INUM = INUM + 1
KVAR(M) = INUM
95 CONTINUE
BUFFER OUT(46,1)(KVAR(1),KVAR(MAXV))
96 IF(UNIT,46)96,97
97 READ (LDICT,3) ((NACON(K,M),M=1,2), K=1,NCOUN)
DO 100 M=1,5
100 CALL SKIPFILE (LDICT)
DO 85 K=1,NCOUN
85 READ (LDICT,4) (BLOC(K,M),M=1,MAXB)

```

```

C
C READ DATA/SALIENCES AND BETAS, CALCULATE ESTIMATES, RESIDUALS AND MEDIANS---
C CREATE ISSUE
C

```

```

II = 0
DO 155 I=1,NVAR
DO 105 M=1,MAXB
105 BETA (M) = 0.
READ 12, NBETA, RANKS
DO 110 M=1,NBETA
READ 5,NUMB,VBETA
110 BETA(NUMB) = VBETA
PRINT 6, ((M, BETA(M)), M=1,MAXB)
115 BUFFER IN(LUIN,1)(ISSUE(1,1,I),ISSUE(NCOUN,1,I))
116 IF(UNIT,LUIN)116,117
117 BUFFER IN(LUIN,1)(ISSUE(1,2,I),ISSUE(NCOUN,2,I))
118 IF(UNIT,LUIN)118,120
120 II=II+1
IF(KVAR(II).LE.0) GO TO 115
IF(NOMED.NE.1) PRINT 8,II
MRESID = 0.

```

```

DO 130 K=1,NCOUN
EST = 0. $ DATA = ISSUE (K,1,I)
IF (DATA .NE. 0.) NOBS(I) = NOBS(I) + 1
DO 125 M=1,MAXB
125 EST = EST + (BLOC(K,M) * BETA(M))
ISSUE (K,1,I) = ((DATA*ISSUE(K,2,I)) + EST)/(ISSUE(K,2,I) +1.)
IF(NOMED.EQ.1) GO TO 126
RESID=(DATA-EST)
IF (DATA .EQ.0.) RESID = 0.
PRINT 9, (NACON(K,KK),KK=1,2),DATA,EST,(ISSUE(K,M,I),M=1,2),RESID
IF(RESID.LT.0.) RESID=RESID*(-1.)
MRESID = MRESID + RESID
126 IF (EST .LT. 1) EST = 1.
IF (ISSUE (K,1,I) .LT. 1.) ISSUE (K,1,I) = 1.
IF (EST .GT. RANKS) EST = RANKS
IF (ISSUE (K,1,I) .GT. RANKS) ISSUE (K,1,I) = RANKS
IF(NOMED.EQ.1) GO TO 130
IF(DATA.NE.0.) WRITE(1)DATA
WRITE(2) EST
WRITE(3)ISSUE(K,1,I)
130 CONTINUE
IF(NOMED.EQ.1) GO TO 155
DO 135 M=1,3
ENDFILE M
135 REWIND M
DO 150 M=1,3
NCOUNT = NCOUN
IF (M .EQ. 1) NCOUNT = NOBS(I)
DO 137 K=1,NCOUNT
137 READ(M) SCORE(K)
DO 145 KK=1,NCOUNT
CURMIN(M)=+999. $NUMC=0
DO 140 K=1,NCOUNT
IF (SCORE (K) .GT. CURMIN (M) ) GO TO 140
CURMIN (M) = SCORE (K)
NUMC=K
140 CONTINUE
SCORE(NUMC)= +999.
IF (KK .NE. ((NCOUNT/2)+1)) GO TO 145
MEDIAN(M) = CURMIN(M)
GO TO 150
145 CONTINUE
150 REWIND M
COUNT = NOBS (I)
MRESID = MRESID / COUNT
PRINT 11,(MEDIAN(M),M=1,3),MRESID
155 CONTINUE
C
C BEST WORST ANALYSIS
C

```

```

159 DO 175 INDEX=1,NRANGE
    READ 12, INUM,MIN,MAX
    IF (KVAR(INUM) .EQ. 0) GO TO 175
    IF (MIN .EQ. 0.) PRINT 13, MAX,INUM
    IF (MAX .EQ. 0.) PRINT 14,MIN,INUM
    IF (MIN .NE. 0. .AND. MAX .NE. 0.) PRINT 15,MIN,MAX,INUM
    I= KVAR(INUM) $ NCOUNT = 0
    DO 170 K=1,NCOUN
        IF (MIN .EQ. 0. .AND. ISSUE(K,1,I) .GT. MAX) GO TO 170
        IF (MAX .EQ. 0. .AND. ISSUE(K,1,I) .LT. MIN) GO TO 170
        IF (MIN .NE. 0. .AND. MAX .NE. 0.) 160,165
    160 IF (ISSUE(K,1,I) .GT. MAX .OR. ISSUE (K,1,I) .LT. MIN) GO TO 170
    165 NCOUNT = NCOUNT + 1
        PRINT 21, (NACON(K,M), M=1,2), ISSUE(K,1,I)
    170 CONTINUE
        PRINT 16, NCOUNT
    175 CONTINUE

```

C
CALCULATE NORMALIZED PREFERENCES AND PACKAGING SALIENCES

```

C
    PRINT 25
    DO 185 I=1,NVAR
        MEAN (I) = 0. $ STD(I) = 0.
        DO 180 K=1,NCOUN
            MEAN (I) = MEAN(I) + ISSUE(K,1,I)
    180 STD(I) = STD(I) + (ISSUE(K,1,I)*ISSUE(K,1,I))
            COUN = NCOUN
            STD(I)=SQRT((STD(I)-((MEAN(I)*MEAN(I))/COUN))/(COUN-1.))
    185 MEAN(I) = MEAN(I)/COUN
        DO 190 K=1,NCOUN
            DO 189 I=1,NVAR
                ISSUE (K,1,I) = (ISSUE (K,1,I)-MEAN(I))/STD(I)
    189 ISSUE(K,2,I) = ISSUE (K,2,I) + 1.
    190 PRINT 17, (NACON(K,M),M=1,2), ((ISSUE(K,MM,I), MM=1,2),I=1,NVAR)
        PRINT 24
        PRINT 18, ((I,MEAN(I), STD(I)),I=1,NVAR)
        BUFFER OUT(46,1)(NVAR,NCOUN)
    191 IF(UNIT,46)191,192
    192 BUFFER OUT(46,1)(MEAN(1),STD(NVAR))
    193 IF(UNIT,46)193,194
    194 DO 199 I=1,NVAR
        BUFFER OUT(46,1)(ISSUE(1,1,I),ISSUE(NCOUN,2,I))
    198 IF(UNIT,46)198,199
    199 CONTINUE

```

C
C SINGLE PACKAGES

```

C
    DO 230 INDEX =1,NSINGLE
        READ 12, INUM,ONE,TWO
        IF(KVAR (INUM) .EQ. 0) GO TO 230

```

```

PRINT 19, INUM, ONE, TWO
I=KVAR(INUM)
ONENOR = (ONE-MEAN(I))/STD(I)
TWNOR=(TWO-MEAN(I))/STD(I)
DO 195 M=1,3
195 VOTE (M) = 0
DO 215 K=1,NCOUN
SCORE(K) = ((ONENOR-ISSUE(K,1,I)) * (ONENOR-ISSUE(K,1,I)))-((TWNOR
1R-ISSUE(K,1,I)) * (TWNOR-ISSUE(K,1,I)))
IF (SCORE(K)) 200,205,210
200 VOTE(1) = VOTE(1) + 1
GO TO 215
205 VOTE (2) = VOTE(2) + 1
GO TO 215
210 VOTE (3) = VOTE (3) + 1
215 CONTINUE
PRINT 20, VOTE(1), ONE, VOTE(2), VOTE(3), TWO
IF(NOLIST.EQ.1) GO TO 230
DO 225 KK=1,NCOUN
CURMAX = -999.
NUMC = 0
DO 220 K=1,NCOUN
IF (SCORE(K) .LT. CURMAX) GO TO 220
CURMAX = SCORE(K) $ NUMC = K
220 CONTINUE
PRINT 21, (NACON(NUMC,M),M=1,2), CURMAX
225 SCORE (NUMC) = -999.
230 CONTINUE

```

C
C
C

FORMAT STATEMENTS

```

1 FORMAT(I2,2I3,4I2,I3,I1)
2 FORMAT (80I1)
3 FORMAT(5X,A8,A6)
4 FORMAT(F4.0,34F3.1,////////)
5 FORMAT (6X,I2,E12.4)
6 FORMAT (9 (I3,F10.1))
7 FORMAT(20F6.2)
8 FORMAT(1H1,21X,*REAL ESTIMATE COMBINED SS RESIDUAL ON
1 VARIABLE*,I5,/)
9 FORMAT(3X,A8,A6,3F10.3,F8.0,F11.3)
11 FORMAT(//,2X,*MEDIAN*,8X,3F10.3,F11.3)
12 FORMAT(I2,2F5.0)
13 FORMAT(1H1,* THE FOLLOWING COUNTRIES HAVE SCORES BELOW*, F4.1,* ON
1VARIABLE*,I5,/)
14 FORMAT(1H1,* THE FOLLOWING COUNTRIES HAVE SCORES ABOVE*, F4.1,* ON
1VARIABLE*,I5,/)
15 FORMAT(1H1,* THE FOLLOWING COUNTRIES HAVE SCORES BETWEEN*,F4.1,*AN
10*,F4.1,* ON VARIABLE*,I5,/)
16 FORMAT(//,2X,I5,* COUNTRIES ARE LISTED ABOVE*)

```



```

17 FORMAT(2X,A8,A7,(10(F6.2,F4.0)) )
18 FORMAT(I5,2F8.2)
19 FORMAT(1X,-----*,//,* ON VARIABLE*,I5,* A COM
  2PARISON BETWEEN RANK*,F5.1,* AND RANK*,F5.1,/)
20 FORMAT(I5,* COUNTRIES PREFER *,F5.0,/,I5,* COUNTRIES HAVE EQUAL PR
  2REFERENCE*,/,I5,* COUNTRIES PREFER *,F5.0,///)
21 FORMAT(3X,A8,A6,F10.3)
24 FORMAT(1H1,* THE MEANS AND STANDARD DEVIATIONS ARE*,//)
25 FORMAT (1H1,* THE NORMALIZED VALUES ARE*,/)
  END

```

PROGRAM MAXPAC

Purpose

This program and its subroutine perform two functions. First, they forecast outcomes on packages of issues, by providing a tally of the number of countries preferring each of the two input packages and a measure of how intense states' preferences are. Second, they produce a series of packages that are more popular than the input package, to the voting body as a whole and to the United States in particular.

Method

After the main problem card is read, the normalized tape from VMODEL is buffered in. If unclassified frequencies of mention are to be used, the national scores tape from SCORES is buffered in, and if multiply used themes are involved, the second file is buffered in from the national scores tape. Then the country dictionary is read and the scores and frequencies of all countries on the issue-variables being processed are printed.

Then, a set of packages at a time, the program reads the issue-variables it includes and the positions of both packages on these issues. These positions are normalized and subroutine PREFER is called.

After the epsilons have been read, the program turns to its second function, producing more popular packages than the one input. It processes one such package at a time, normalizing it and then calculating the normalized direction cosine of each country on each issue in the package. After a composite direction cosine has been calculated for each issue, the program generates a series of packages that move away from the input package in the direction calculated. The series is produced by moving in that direction at different distances (epsilons). For each epsilon tested, subroutine PREFER is called.

Subroutine PREFER calculates a utility score for each state for the two packages being compared. The sign of the score determines which package the state prefers and the magnitude determines the intensity of its choice. The results are printed, if requested, so are the utility scores (in decreasing order).

Dimensioned Variables

There are 15 dimensioned variables in this program and its subroutine:

(1) KVAR has 70 fields, one for each existing issue-variable number. Each contains a flag for the issue-variables on the input tape; i.e., $KVAR(N) = J$ when issue-variable N is the J th one on the input tape and is 0 when N is not on the tape.

(2) MEAN, dimensioned 45, contains the means of the 45 possible issue-variables on the input tape.

(3) STD, dimensioned 45, contains the standard deviations of the 45 possible issue-variables on the input tape.

(4) ISSUE, a three dimensional matrix 149 by 2 by 45, stores the standardized preferred position and frequency of mention of each country on each of the 45 possible issue-variables.

(5) NACON, dimensioned 149 by 2, serves two functions. It is temporary storage of the unclassified national score and frequency of mention for each country on the variable being processed. During most of the program, it contains a two field name for each country.

(6) DOUBL is dimensioned 149 by 10. A row represents a country and a column represents one of the 10 possible multiply used themes. The matrix contains the number of times a particular country has said one of these themes.

(7) WEIGHT, dimensioned 10, contains the proportion of mentions of each of the multiply used themes that is to be assigned to the issue-variable being processed.

(8) PACK, dimensioned 45 by 2, contains the positions on the 45 possible issue-variables for the two packages being processed.

(9) PVAR, dimensioned 45, stores the numbers for the 45 possible issue-variables in the packages being processed.

(10) VOTE, an array dimensioned 4, contains the number of states that (1) strongly prefer package 1, (2) marginally prefer package 1, (3) marginally prefer package 2, and (4) strongly prefer package 2.

(11) SCORE, a matrix dimensioned 149 by 3, contains the utility score for package 1, that for package 2, and the difference between the two, for each country.

(12) THETA, dimensioned 45, contains the composite direction cosines for the 45 possible issue-variables in the package being maximized.

(13) Z, a matrix 45 by 149, stores the direction cosine for each country on each of the 45 possible issue-variables in the package being maximized.

(14) ESPLN is an array containing up to 16 user specified intensity weights which in combination with THETA produce various maximized packages.

(15) NUM, dimensioned 45, stores the direction cosines for the 45 possible issue-variables in the package being maximized for the United States.

Operational Effects of System Changes

This program is dependent upon the output tape from VMODEL, the country dictionary, and possibly the national scores tape from SCORES. If their structure and format remain the same, the following modifications to them can be handled by changing the following variables:

If the number of countries exceeds 149, ISSUE, NACON, DOUBL, SCORE, and Z will have to be redimensioned.

If the number of existing issue-variables ever exceeds 70, KVAR, MEAN, and STD will have to be redimensioned and MAXV will also have to be increased.

If the number of issue-variables being processed through VMODEL exceeds 45, ISSUE, PACK, THETA, NUM, Z, and PVAR will have to be redimensioned.

If more than 10 multiply used themes are processed through SCORES, DOUBL and WEIGHT will have to be redimensioned.

If more than 16 intensity weights are necessary, ESPLN will have to be increased.

Program Listing

The program and subroutine listings follow.

```

PROGRAM MAXPAC
COMMON VOTE(4), SCORE(149,3), PVAR(45), PACK(45,2), KVAR(70), ISSUE(149
1,2,45), NACON(149,2), MEAN(45), STD(45), NVAR, NCOUN
DIMENSION THETA(45), NUM(45), Z(45,149), ESPLN(16), DOUBL(149,10), WEIG
1HT(10)
TYPE REAL MEAN, ISSUE, NACON, NUM
TYPE INTEGER PVAR
MAXV = 70

C
C READ MAIN PROBLEM CARD AND NORMALIZED TAPE
C
READ 1, LUIN, LDICT, LUS, LDOUB, NDOUB, LIST, NPACK, NUS, NMAXP, NEP
1S
BUFFER IN(LUIN,1)(KVAR(1),KVAR(MAXV))
30 IF(UNIT,LUIN)30,31
31 BUFFER IN(LUIN,1)(NVAR,NCOUN)
32 IF(UNIT,LUIN)32,33
33 NVAR=NVAR
BUFFER IN(LUIN,1)(MEAN(1),STD(NVAR))
34 IF(UNIT,LUIN)34,35
35 DO 38 I=1,NVAR
NCOUN=NCOUN
BUFFER IN(LUIN,1)(ISSUE(1,1,I),ISSUE(NCOUN,2,I))
36 IF(UNIT,LUIN)36,38
38 CONTINUE

C
C IF UNCLASSIFIED SALIENCES ARE TO BE USED, READ THEM AND SUBSTITUTE
C FOR EXISTING SALIENCES
C
IF (LUS .EQ. 0) GO TO 100
DO 50 I=1,NVAR
BUFFER IN(LUS,1)(NACON(1,1),NACON(NCOUN,1))
39 IF(UNIT,LUS)39,40
40 BUFFER IN(LUS,1)(NACON(1,2),NACON(NCOUN,2))
41 IF(UNIT,LUS)41,42
42 DO 45 K=1,NCOUN
45 ISSUE(K,2,I)=NACON(K,2)
50 CONTINUE

C
C IF MULTIPLY USED THEMES ARE USED, READ AND CALCULATE NEW SALIENCE
C
100 IF (LDOUB .EQ. 0) GO TO 150
CALL SKIPFILE (LDOUB)
BUFFER IN(LDOUB,1)(DOUBL(1,1),DOUBL(NCOUN,NDOUB))
101 IF(UNIT,LDOUB)101,102
102 DO 110 I=1,NVAR
READ 8, (WEIGHT(M), M=1,NDOUB)
DO 109 K=1,NCOUN
DO 105 M=1,NDOUB
IF (WEIGHT(M) .GE. 0.) ISSUE (K,2,I) = ISSUE (K,2,I) -
2DOUBL (K,M) + (WEIGHT (M) * DOUBL (K,M) )
105 CONTINUE
109 CONTINUE
110 CONTINUE
C

```

```

C      READ COUNTRY DICTIONARY AND PRINT NATIONAL SCORES AND SALIENCES
C
150 PRINT 2
    DO 120 K=1,NCOUN
      READ (LDICT,9) (NACON (K,M), M=1,2)
      PRINT 10, NACON(K,1), ((ISSUE(K,M,I), M=1,2), I=1,NVAR)
120 CONTINUE
C
C      PROCESS ONE SET OF PACKAGES AT A TIME
C
    DO 1000 INDEX = 1,NPACK
      READ 11,NOVAR,(PVAR(I), I=1,NVAR)
      READ 12, ((PACK (I,M), I=1,NOVAR), M=1,2)
      DO 184 M=1,4
184 VOTE(M)=0.
      DO 185 M=1,3
      DO 185 K=1,NCOUN
185 SCORE (K,M) =0.
      PRINT 13,(PVAR(I),I=1,NOVAR)
      PRINT 14,(PACK(I,1),I=1,NOVAR)
      PRINT 15,(PACK(I,2),I=1,NOVAR)
      DO 200 I=1,NOVAR
        M = KVAR (PVAR(I) )
        DO 190 MM=1,2
190 PACK (I,MM) = (PACK (I,MM) - MEAN (M)) /STD (M)
200 CONTINUE
        IFLAG=1
        CALL PREFER(IFLAG,LIST,NOVAR,NUS,CEPS)
1000 CONTINUE
C
C      MAXIMIZATION MODEL
C
      READ 20,(ESPLN(M),M=1,NEPS)
      DO 2000 INDEX=1,NMAXP
        READ 11,NOVAR,(PVAR(I),I=1,NVAR)
        READ 12,(PACK(I,1),I=1,NOVAR)
        PRINT 21,(PVAR(I),I=1,NOVAR)
        PRINT 22,(PACK(I,1),I=1,NOVAR)
        DO 1100 I=1,NOVAR
          M=KVAR(PVAR(I))
          PRINT 1101,MEAN(M),STD(M),PVAR(I)
1100 PACK(I,1)=(PACK(I,1)-MEAN(M))/STD(M)
C
C      CALCULATE THE DIRECTION COSINES/NORMALIZE THEM
C
      DO 1200 K=1,NCOUN
      DO 1200 I=1,NOVAR
        M=KVAR(PVAR(I))
1200 Z(I,K)=ISSUE(K,2,M)*((ISSUE(K,1,M)-PACK(I,1))
      DO 1300 K=1,NCOUN
      SUM=0.
      DO 1350 I=1,NOVAR
1350 SUM=SUM+ABS(Z(I,K))
      DO 1360 I=1,NOVAR
1360 Z(I,K)=Z(I,K)/SUM

```

```

1300 CONTINUE
      PRINT 18
      DO 1370 K=1,NCOUN
      PRINT 19,NAON(K,1),(Z(I,K),I=1,NOVAR)
1370 CONTINUE

```

C

C CALCULATE PHI...SUMMED ACROSS COUNTRIES

C

```

      SUMPT=0.
      DO 1500 I=1,NOVAR
      THETA(I)=0.
      DO 1400 K=1,NCOUN
1400 THETA(I)=THETA(I)+Z(I,K)
1500 SUMPT=SUMPT+(THETA(I)*THETA(I))

```

C

C CALCULATE THETA AND OPTIMUM PACKAGE

C

```

      DO 1600 I=1,NOVAR
1600 THETA(I)=THETA(I)/SQRT(SUMPT)
      PRINT 23,(THETA(I),I=1,NOVAR)
      IFLAG=2
      DO 1950 IJK=1,NEPS
      CEPS=ESPLN(IJK)
      DO 1650 I=1,NOVAR
1650 PACK(I,2)=PACK(I,1)+(ESPLN(IJK)*THETA(I))
      DO 1655 M=1,4
1655 VOTE(M)=0.
      DO 1660 M=1,3
      DO 1660 K=1,NCOUN
1660 SCORE(K,M)=0.
      CALL PREFER(IFLAG,LIST,NOVAR,NUS,CEPS)
1950 CONTINUE
      IF(IFLAG.NE.0) GO TO 2000
      X=0. $WS=0. $DEM=0.
      DO 1960 I=1,NOVAR
      NUM(I)= THETA(I)* SQRT(SUMPT)
      X=X+(NUM(I)*Z(I,NUS))
1960 WS=WS+(Z(I,NUS)*Z(I,NUS))
      IF(X.GE.0.) X=0.
      DO 1970 I=1,NOVAR
      NUM(I)=NUM(I)-((Z(I,NUS)*X)/WS)
1970 DEM=DEM+(NUM(I)*NUM(I))
      DO 1980 I=1,NOVAR
1980 THETA(I)=NUM(I)/SQRT(DEM)
      PRINT 3001,(THETA(I),I=1,NOVAR)
2000 CONTINUE

```

C

C FORMAT STATEMENTS

C

```

1 FORMAT (20I3)
2 FORMAT (/,* NORMALIZED PREFERRED POSITIONS AND SALIENCES*,/)
8 FORMAT (16F5.1)
9 FORMAT (5X, A8, A6)
10 FORMAT (X,A8,12 (F6.2,F4.0), /, 8X, 12(F6.2, F4.0), /, 8X, 12(F6.2,
2F4.0), /, 4(F6.2, F4.0))

```

```

5 FORMAT (12F10.3)
11 FORMAT (40I2)
12 FORMAT (40F2.0)
13 FORMAT (//,* -----*,/,* VARIABLES IN PACKAGES*,/, (13X(I3,19I6))
1)
14 FORMAT(* PACKAGE 1=*,(20F6.3))
15 FORMAT(* PACKAGE 2=*,(20F6.3))
18 FORMAT(1H1,* THE DIRECTION COSINES ARE*/ )
19 FORMAT(1X,A8,(14F9.3))
20 FORMAT(16F5.0)
21 FORMAT(1H1,* BASE PACKAGE CONTAINS*,/,* VARIABLES*,5X,(40I3))
22 FORMAT(* SCALE VALUES*,2X,(40F3.0))
23 FORMAT(1X,* THE THETA VALUES ARE*,/, (14F8.3))
26 FORMAT(20F5.2)
1101 FORMAT(* WHERE THE MEAN IS *,F5.2,* STD IS *,F5.2,* FOR VAR *,I4)
3001 FORMAT(1H1,* THE US PACKAGE THAT WE WOULD PREFER IS*,/, (14F9.5))
END

```



```

SUBROUTINE PREFER(IFLAG,LIST,NOVAR,NUS,CEPS)
COMMON VOTE(4),SCORE(149,3),PVAR(45),PACK(45,2),KVAR(70),ISSUE(149
1,2,45),NACON(149,2),MEAN(45),STD(45),NVAR,NCOUN
TYPE REAL MEAN, ISSUE, NACON, NUM
TYPE INTEGER PVAR

```

```

THIS SUBROUTINE DETERMINES WHICH PACKAGE EACH COUNTRY PREFERS
IF REQUESTED, IT PRINTS A LIST OF COUNTRIES THAT PREFER EACH
PACKAGE

```

```

DO 300 K=1,NCOUN
DO 250 I = 1,NOVAR
M = KVAR (PVAR (I))
DO 210 MM = 1,2
210 SCORE (K,MM) = SCORE (K,MM) + (ISSUE (K,2,M)
2 * ((ISSUE (K,1,M) - PACK(I,MM)) * (ISSUE (K,1,M) -PACK(I,MM))))
250 SCORE (K,3) = SCORE (K,3) + ISSUE (K,2,M)
SCORE (K,3) =(SCORE (K,1) - SCORE (K,2))/ SCORE(K,3)
IF (SCORE (K,3) .LE. 1. .AND. SCORE (K,3) .GE. 0.) GO TO 280.
IF (SCORE (K,3) .GE. -1. .AND. SCORE (K,3) .LE. 0.) GO TO 270
IF (SCORE (K,3)) 260,270,290
260 VOTE(1) = VOTE(1) + 1.
GO TO 300
270 VOTE (2) = VOTE(2) + 1.
GO TO 300
280 VOTE(3) = VOTE(3) + 1.
GO TO 300
290 VOTE (4) = VOTE (4) + 1.
300 CONTINUE
IF(IFLAG.EQ.1) 350,325
325 PRINT 3,CEPS
DO 330 I=1,NOVAR
M=KVAR(PVAR(I))
A=MEAN(M)+(PACK(I,1)*STD(M))
B=MEAN(M)+(PACK(I,2)*STD(M))
330 PRINT 1749,PVAR(I),A,B
PRINT 25,SCORE(NUS,3)
IF(SCORE(NUS,3).LT.0.) IFLAG=0
350 PRINT 16,(VOTE(M),M=1,4)
IF(LIST.EQ.0) GO TO 1000
DO 500 K=1,NCOUN
CURMAX = -9999. $ NCOUNT = 0
DO 450 KK=1,NCOUN
IF (SCORE(KK,3) .LT. CURMAX) GO TO 450
CURMAX = SCORE(KK,3) $ NCOUNT = KK
450 CONTINUE
PRINT 17,(NACON(NCOUNT,M),M=1,2),SCORE(NCOUNT,3)
SCORE(NCOUNT,3)=-9999.
500 CONTINUE

```

```

FORMAT STATEMENTS

```

```

3 FORMAT(//,* -----*,/,* PACKAGING ANALYSIS CONTAINS THE FOLLOWI
1NG COMPONENTS WHEN EPSILON IS *,F6.2,//,* VAR NO. BASE PACKAGE
1 MODEL PACKAGE*/)

```

```

16 FORMAT(//F5.0,* COUNTRIES PREFER PACKAGE 1*/F5.0,* COUNTRIES MARGI
  1NALLY PREFER PACKAGE 1*,/,F5.0,* COUNTRIES marginally prefer packa
  2GE 2*,/,F5.0,* COUNTRIES PREFER PACKAGE 2*,//)
17 FORMAT(X,2A8,F10.2)
25 FORMAT(//,* THE US UTILITY ON THESE PACKAGES IS*,F10.5,/)
1749 FORMAT(I5,2F16.5)
1000 RETURN
      END

```

PROGRAM VALID

Purpose

This program calculates a crude measure of the degree of similarity between CNA's forecasts and 'outside' forecasts. Comparisons can be performed on 45 issue-variables in a single run.

Method

After the normalized tape (output from VMODEL) has been buffered in, the 'outside' forecasts are read and normalized. For each issue being processed, states' preferred positions are compared with the 'outside' forecast and the number of countries with preferred positions less than the forecast is calculated. This is converted to a percentage and a crude similarity measure calculated through the following formula:

$$\left| \log \left(\frac{\text{percent}}{1-\text{percent}} \right) \right|$$

This measure is printed and the next issue processed.

Dimensioned Variables

This program contains five dimensioned variables:

- (1) MEAN, dimensioned 45, contains the means of the 45 possible issue-variables on the input tape.
- (2) STD, dimensioned 45, contains the standard deviations of the 45 possible issue-variables on the input tape.
- (3) ISSUE, a three dimensional matrix 149 by 2 by 45, contains the preferred position and frequency of mention for each country on each of the 45 possible issue-variables.
- (4) KVAR has 70 fields, one for each existing issue-variable number. It contains counters for the issue-variables included on the input tape; i.e., $KVAR(N) = J$ when issue-variable N is the Jth issue-variable listed on the tape and is 0 if N is not on the tape.
- (5) PVAR, dimensioned 45, contains the 'outside' forecast for each of the 45 possible issue-variables on the input tape.

Operational Effects of System Changes

This program depends entirely upon the structure of the binary tape produced by VMODEL. If any changes are made to it, the four read statements must be changed. If these changes increase the number of countries, processed issue-variables, or existing issue-variables, the following dimensions must be changed:

If the number of countries exceeds 149, ISSUE will have to be enlarged.

If the number of issue-variables being processed through VMODEL exceeds 45, MEAN, STD, ISSUE, and PVAR will have to be redimensioned.

If the number of existing issue-variables exceeds 70, MAXV will have to be increased and KVAR redimensioned.

Program Listing

The listing of program VALID follows.

```

PROGRAM VALID
COMMON MEAN(45),STD(45),ISSUE(149,2,45),PVAR(45),KVAR(70)
COMMON NVAR,NCCUN
TYPE REAL ISSUE,MEAN

```

C
C
C
C

THIS PROGRAM PROVIDES A CRUDE MEASURE OF THE DEGREE OF
SIMILARITY BETWEEN OUR FORECASTS AND OUTSIDE FORECASTS

```

MAXV=70 $LUIN=1
BUFFER IN(LUIN,1)(KVAR(1),KVAR(MAXV))
10 IF(UNIT,LUIN) 10,20
20 BUFFER IN(LUIN,1)(NVAR,NCCUN)
30 IF(UNIT,LUIN) 30,40
40 BUFFER IN(LUIN,1)(MEAN(1),STD(NVAR))
50 IF(UNIT,LUIN) 50,60
60 DO 90 I=1,NVAR
    BUFFER IN(LUIN,1)(ISSUE(1,1,I),ISSUE(NCCUN,2,I))
85 IF(UNIT,LUIN) 85,90
90 CONTINUE
    NVAR=NVAR
    READ 6,(PVAR(I),I=1,NVAR)
    DO 98 I=1,NVAR
        DO 95 K=I,MAXV
            IF(KVAR(K).LE.0) 95,96
95 CONTINUE
96 KVAR(K)=0
98 KVAR(I)=K
    PRINT 7
    DO 100 I=1,NVAR
        IF(PVAR(I).LE.0.) GO TO 100
        PVAR(I)=(FVAR(I)-MEAN(I))/STD(I)
        COUNT=0.
        DO 200 K=1,NCCUN
            IF(ISSUE(K,1,I).LT.PVAR(I))COUNT=COUNT+1.
200 CONTINUE
        COUNT=COUNT/149.
        VALUE=ABS(LOGF(CCOUNT/(1.-COUNT)))
        PVAR(I)=(PVAR(I)*STD(I))+MEAN(I)
        PRINT 8,KVAR(I),PVAR(I),COUNT,VALUE
100 CONTINUE

```

C
C
C

FORMAT STATEMENTS

```

6 FORMAT(40F2.0)
7 FORMAT(/,* VARIABLE*,3X,*OUTSIDE*,6X,*PERCENT*,6X,*SIMILIARITY*,/,
14X,*NO.*,4X,*FORECAST*,2X,*BELOW FO<ECAST*,5X,*MEASURE*,//)
8 FORMAT(4X,I3,4X,F6.1,8X,F5.3,10X,F7.4)
END

```

APPENDIX A

PROGRAMS AFFECTED BY
CHANGES TO DICTIONARY AND DATA TAPES

APPENDIX A

PROGRAMS AFFECTED BY CHANGES TO DICTIONARY AND DATA TAPES

This appendix provides the programmer with a cross-reference guide that summarizes the effect of changes to the dictionary and data tapes on all programs in the system. It should be used whenever there is a change to the structure or format of these tapes or whenever the dictionary tape is expanded.

Table A-1 identifies the dictionary files that are used by each of the programs. When changes are made to any of these files, the programmer should refer to the section "Operational Effects of System Changes" for each of the programs checked.

Table A-2 identifies the data tapes that are used or created by each of the programs. When changes are made to any of these tapes, the programmer should refer to the sections "Method" and "Dimensioned Variables" for each of the programs checked.

In addition there are five possible areas of expansion that affect more than one program in the system:

If the number of time periods included in an analysis ever exceeds 4, programs MATRIX, PROFLE, PTHEME, and SCORES will have to be changed.

If the number of documents from which raw data is extracted exceeds 999, programs MATRIX and LETTER will have to be changed.

If the structure of an update card changes, programs CHECK and UPDATE will have to be changed.

If more than 45 issue-variables need to be processed in a single run, programs MAXPAC, MBLOC, PROFLE, SCORES, and VMODEL will have to be changed.

If two or more variables to which more than 10 multiply used themes have been assigned in common need to be processed in a single run, programs SCORES and MAXPAC will have to be changed.

TABLE A-1

PROGRAMS AFFECTED BY CHANGES TO DICTIONARY FILES

<u>Program</u>	<u>Country</u>	<u>Theme</u>	<u>Dictionary file</u>			
			<u>Major/ minor</u>	<u>Issue- variable</u>	<u>Lettered theme</u>	<u>Independent variable</u>
BIMED						X
CHECK	X					
CODEBK	X	X	X	X		
DTAPE	X	X	X	X	X	X
LETTER	X	X				
MATRIX	X	X	X			
MAXPAC	X			X		
MBLOC	X					X
PROFLE	X	X	X	X		
PTHEME	X	X	X	X		
REMOVE		X				
SCORES	X	X		X		X
SEARCH		X	X	X		
UPDATE						
VALID	X			X		
VMODEL	X			X		X

TABLE A-2
PROGRAMS AFFECTED BY CHANGES TO DATA TAPES

<u>Program</u>	<u>Data tape</u>					
	<u>Raw data</u>	<u>Matrix</u>	<u>Issue-variable</u>	<u>National scores</u>	<u>Regression input</u>	<u>Normalized variable</u>
BIMED					X	
CHECK						
CODEBK						
DTAPE						
LETTER	X					
MATRIX	X	X*				
MAXPAC				X		X
MBLOC				X		
PROFLE		X	X*			
P THEME		X				
REMOVE	X					
SCORES			X	X*	X*	
SEARCH						
UPDATE	X*					
VALID						X
VMODEL				X		X*

*Tapes were created by these programs.

END

FILMED

9-83

DTIC